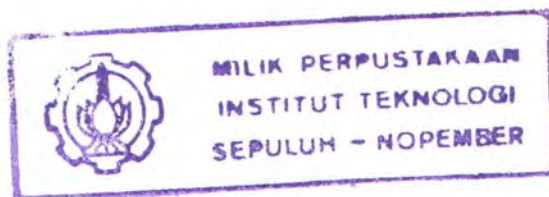


22308/11/05



# PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK OLAP BUILDER DENGAN MENGGUNAKAN ORACLE OLAP DAN ORACLE BUSINESS INTELLIGENCE

## TUGAS AKHIR



RSIF  
005.1  
Kun  
P-1  
2004

Disusun Oleh :

ANANG KUNAEFI  
5198100048

PERPUSTAKAAN ITS	
Tgl. Terima	
Terima Dari	
No. Agenda Prp.	

JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2004

**PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK  
OLAP BUILDER DENGAN MENGGUNAKAN ORACLE OLAP  
DAN ORACLE BUSINESS INTELLIGENCE**

**TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Komputer  
Pada  
Jurusan Teknik Informatika  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Surabaya**

**Mengetahui/Menyetujui**



**Dosen Pembimbing**

*[Handwritten signature]*

*20/07/04*

**Rully Soelaiman, S.Kom, M.Kom**  
**NIP 132085802**

**SURABAYA  
JULI 2004**



*BUKU INI KUPERSEMBAHKAN UNTUK  
KEDUA ORANG TUAKU, BAPAK DAN IBU,  
DAN UNTUK SEMUA ORANG YANG PERCAYA  
BAHWA KEBERHASILAN ADALAH BUAH  
DARI KEGIGIHAN DAN KERJA KERAS.  
Faidza azzamta fatawakkal/alallah.*

## ABSTRAK

*Kebutuhan akan penggunaan teknologi OLAP semakin meningkat seiring dengan besarnya manfaat yang didapatkan. Oracle9i sebagai produk database yang banyak digunakan oleh korporasi menyediakan fasilitas untuk mendukung teknologi OLAP ini*

*Tidak seperti versi sebelumnya, pada Oracle9i, Oracle menggabungkan Oracle OLAP ke dalam databasenya. Integrasi ini dimaksudkan untuk mendukung proses pengolahan data OLAP sebagai sarana pendukung keputusan dalam proses bisnis perusahaan. Namun, penggunaan teknologi OLAP pada Oracle ini masih dirasa terlalu sulit dan memiliki kelemahan. Kesulitan terletak pada proses pembentukan metadata OLAP. Kelemahan terletak pada validasi OLAP object (dimensi, cube dan measure) serta prosedur pemanfaatan metadata tersebut.*

*Untuk itulah, dalam tugas akhir ini dibuat sebuah tool untuk memanfaatkan teknologi OLAP pada Oracle secara lebih mudah. Tool ini diperuntukkan untuk proses pembentukan metadata OLAP dan pemanfaatannya dalam bentuk analisis multidimensi.*

*Uji coba dilakukan dengan melakukan pembuatan metadata OLAP pada data warehouse Foodmart yang merupakan suatu sistem database yang menangani penyimpanan dan pengelolaan informasi usaha pemasaran produk-produk makanan. Selanjutnya, metadata tersebut ditampilkan dalam bentuk data multidimensi. Hasil uji coba perangkat lunak menunjukkan kemampuan dalam pembentukan metadata OLAP dengan lebih mudah serta menampilkan dan mengolah data multidimensi dalam bentuk crosstab dan grafik.*

**Kata Kunci :** *Data warehouse, Oracle OLAP, Oracle Business Intelligence Beans.*





## **KATA PENGANTAR**

Segala puji bagi Allah yang telah memberikan karunia-Nya sehingga penulis bisa menyelesaikan tugas akhir yang berjudul :

### **“PERANCANGAN DAN PEMBUATAN PERANGKAT LUNAK OLAP BUILDER DENGAN MENGGUNAKAN ORACLE OLAP DAN ORACLE BUSINESS INTELLIGENCE”**

Pengerjaan tugas akhir ini, bagi penulis, merupakan suatu kesempatan yang sangat baik untuk mengimplementasikan dan lebih memperdalam semua pengetahuan yang didapatkan penulis selama di perkuliahan.

Penulis berharap, apa yang ditulis di dalam buku tugas akhir ini dapat menjadi bagian dari pengembangan teknologi informasi saat ini. Serta, dapat memberikan kontribusi bagi kampus Teknik Informatika tercinta secara khusus dan masyarakat TI secara umum.

Akhirnya, semoga Allah mengijinkan penulis untuk dapat lebih banyak berkiprah dalam dunia teknologi informasi setelah ini.

Surabaya, Juli 2004

**Penulis**

## UCAPAN TERIMA KASIH

Segala puji syukur kehadiran Allah SWT yang memberikan karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir ini. Sholawat dan salam semoga senantiasa tercurahkan bagi Nabi Muhammad SAW beserta sahabat dan keluarganya.

Selain itu, penulis mengucapkan terima kasih yang sebesar-besarnya kepada mereka yang telah membantu penulis dalam menyelesaikan tugas ini, yaitu:

1. Kedua orang tua penulis, bapak dan ibu, atas doa dan dorongannya sehingga penulis bisa terus termotivasi dalam menyelesaikan tugas akhir ini.
2. Dosen pembimbing, Bapak Rully Soelaiman, atas segala bimbingan dan bantuan selama pengerjaan tugas. Penulis merasa sangat terbantu pada masa-masa persiapan ujian sidang dengan fasilitas yang diberikan pembimbing. Komputer dan printer pinjamannya benar-benar sangat cepat, Pak!.
3. Ketua Jurusan Teknik Informatika ITS, Bapak Yudhi Purwananto dan Bapak Dwi Sunaryono selaku dosen wali dari penulis.
4. Para dosen penguji, Bapak Djoko Lianto, Bu Chastine, dan Pak Darlis yang memberikan pertanyaan-pertanyaan bermutu selama proses ujian, sehingga memberikan kesempatan kepada penulis untuk menjelaskan dengan lebih baik.
5. Keluargaku yang mendukung, Mas Affan, Mbak Phat, Mas Joko, Cacek, Mbak Erlik dan lain-lain. Dan juga tak lupa buat si kecil Akmal yang nggak pernah berhenti ‘ngoceh’, besok ammi belikan es krim lagi.



6. Teman-teman seperjuangan, Gus Mul, Gus Pras, Decki, Ary Shidiq, Budi, Wawan, Hendra, Saenal, Ridha, Aak, Dhane, Adhit, Faida, Mas Jatun, Okky Akbar dan lain-lain.
7. Teman-teman admin dan penghuni Laboratorium Sistem Informasi yang membantu memberikan fasilitas tambahan bagi penulis, Wahib, Agung, Fauzan, Bunug, Ary dan Udin.
8. Teman-teman yang jauh dimata dekat di monitor (chatting), Bhakti, Kurnix, Iqbal, Andhank, Darojat, Maya, Rona, Arief Sungkar, Istas dan Novek.
9. Teman-teman segroup yang begitu pengertian, Pak Dedi, Pak Hasan, Pak Gaguk, Pak Andriy, Pak Dipo, Pak Eddy, Pak Heri, Pak Yoyon dan Pak Husni. Maapin, soalnya banyak melimpahkan tugas-tugas yang seharusnya penulis kerjakan. Terima kasih juga doanya.
10. Adik-adik yang tidak kenal lelah memperjuangkan keadilan bagi tunas-tunas bangsa, Cahyo, Luqman, Eric, Te, Anom, Wahyu, Andra, Detak, Kristiawan, Zakaria dan mbak-mbak yang ada dibelakangnya Atiek, Betty dan Anna.
11. Karyawan TC, mas Hermono, Pak Pri, Mbak Eva dan Mas Yudhi. Terima kasih atas bantuan dan segala kemudahan yang diberikan kepada penulis.
12. Para pejuang keadilan dan kesejahteraan di struktur cabang maupun daerah yang tidak mungkin tersebut satu per satu, yang telah begitu pengertian mendukung penulis dalam setiap langkah menuju kelulusan. Alhamdulillah, kemarin kita menang!. Semoga dapat tiga besar di tahun 2009, Oke!.
13. Aa' Gym, Ust. Anis Matta, Gede Prama dan Forum Lingkar Pena yang ceramah dan bukunya dapat memberikan nuansa baru bagi penulis.

## DAFTAR ISI

ABSTRAK .....	I
KATA PENGANTAR.....	II
UCAPAN TERIMA KASIH .....	III
DAFTAR ISI.....	V
DAFTAR GAMBAR.....	VIII
DAFTAR TABEL .....	X
BAB I PENDAHULUAN.....	1
1.1 LATAR BELAKANG .....	1
1.2 PERMASALAHAN .....	2
1.3 BATASAN PERMASALAHAN.....	3
1.4 TUJUAN.....	4
1.5 METODOLOGI PEMBUATAN TUGAS AKHIR.....	4
1.6 SISTEMATIKA PENULISAN .....	5
BAB II TEORI PENUNJANG.....	7
2.1 DATA WAREHOUSE .....	7
2.2 ON LINE ANALYTICAL PROCESSING (OLAP).....	10
2.2.1 Data Multidimensi.....	11
2.2.2 Operasi Data Multidimensi.....	12
2.2.2.a. Rotation/Pivoting .....	12
2.2.2.b Slice dan Dice .....	13
2.2.2.c Drill Down dan Consolidation .....	14
2.2.3 Skema Data Pada OLAP.....	15
2.2.3.a Star Schema.....	15
2.2.3.b Snowflake Schema.....	16
2.2.4 OLAP Metadata Model.....	17
2.2.4.a Dimensi .....	17
2.2.4.b Cube .....	18
2.2.4.c Hierarchy .....	19
2.2.4.d Level.....	19
2.2.4.e Attribute .....	20
2.2.4.f Tabel Fakta .....	20
2.2.4.g Measure .....	21
2.3 ORACLE9i OLAP .....	21
2.3.1 Oracle9i OLAP API.....	22
2.3.2 OLAP Catalog.....	22
2.3.3 Membuat OLAP Metadata Dengan OEM.....	22
2.4 ORACLE JDBC.....	24
2.5 ORACLE BUSINESS INTELLIGENCE BEANS.....	25



<b>BAB III DESAIN PERANGKAT LUNAK.....</b>	<b>27</b>
3.1 DESAIN PROSES.....	27
3.1.1 <i>Use Case Diagram</i> .....	27
3.1.2 <i>Activity Diagram</i> .....	28
3.1.3 <i>Sequence Diagram</i> .....	32
3.2 DESAIN OBYEK .....	38
3.2.1 <i>Obyek Pada Proses Dimensi</i> .....	39
3.2.2 <i>Obyek Pada Proses Cube</i> .....	41
3.2.3 <i>Obyek Pada Proses Presentasi Data</i> .....	42
3.3 DESAIN ANTARMUKA .....	43
<b>BAB IV IMPLEMENTASI PERANGKAT LUNAK .....</b>	<b>46</b>
4.1 ARSITEKTUR SISTEM.....	46
4.2 TAHAP PERSIAPAN .....	47
4.3 IMPLEMENTASI PROSES .....	49
4.3.1 <i>Pembuatan Koneksi Oracle</i> .....	49
4.3.1.a <i>Koneksi Oracle OLAP</i> .....	49
4.3.1.b <i>Koneksi Oracle BI Beans</i> .....	49
4.3.2 <i>CWM Package Pada Oracle</i> .....	50
4.3.3 <i>Class Operation</i> .....	51
4.3.4 <i>Class HierarchyDim</i> .....	53
4.3.5 <i>Class LevelDim</i> .....	54
4.3.6 <i>Class Relation</i> .....	55
4.3.7 <i>Attribute Long_Description Dan Short_Description</i> .....	55
4.3.8 <i>Metadata Refresh</i> .....	57
4.4 IMPLEMENTASI ANTARMUKA.....	58
4.4.1 <i>Menu Connection</i> .....	59
4.4.2 <i>Menu OLAP</i> .....	61
4.4.3 <i>Menu View</i> .....	62
4.4.4 <i>Menu About</i> .....	63
4.4.5 <i>Context Menu</i> .....	63
4.4.6 <i>Form FrameDimension</i> .....	65
4.4.7 <i>FrameCube</i> .....	70
<b>BAB V UJICOBAN DAN EVALUASI .....</b>	<b>74</b>
5.1 LINGKUNGAN UJI COBA.....	74
5.2 DATABASE UJI COBA .....	74
5.3 EKSTRAKSI DATA.....	75
5.3.1 <i>Konversi File Text</i> .....	75
5.3.2 <i>Membuat Control File</i> .....	76
5.3.3 <i>Loading Data</i> .....	76
5.4 SKENARIO UJI COBA .....	77
5.3.1 <i>Uji Coba 1 : Membuat Metadata OLAP</i> .....	78
5.3.1.A <i>Pembuatan Dimensi Standar</i> .....	78
5.3.1.B <i>Pembuatan Dimensi Waktu</i> .....	81
5.3.1.C <i>Pembuatan Dimensi Snowflake</i> .....	83

5.3.1.D Pembuatan Cube.....	88
5.3.2 Uji Coba 2 : Mengolah Data Multidimensi .....	91
5.2.4.A SALES_CUBE Crosstab .....	91
5.2.4.B SALES_CUBE Grafik.....	96
5.4. EVALUASI .....	97
<b>BAB VI PENUTUP .....</b>	<b>98</b>
6.1 KESIMPULAN.....	98
6.2 SARAN .....	99
<b>DAFTAR PUSTAKA.....</b>	<b>100</b>
<b>LAMPIRAN A.....</b>	<b>101</b>
<b>TABEL KELEMAHAN TEKNOLOGI OLAP PADA ORACLE9I.....</b>	<b>101</b>
<b>LAMPIRAN B .....</b>	<b>103</b>
<b>PROSES INSTALASI PATCH 9.2.0.2.1 PADA ORACLE9I.....</b>	<b>103</b>
<b>LAMPIRAN C .....</b>	<b>107</b>
<b>PROSES INSTALASI BUSINESS INTELLIGENCE BEANS.....</b>	<b>107</b>



## DAFTAR GAMBAR

GAMBAR 2.1 ARSITEKTUR DATA WAREHOUSE .....	9
GAMBAR 2.2 OLAP DIANTARA TEKNOLOGI LAIN .....	10
GAMBAR 2.3 DATA DIMENSI TUNGGAL .....	11
GAMBAR 2.4 DATA MULTIDIMENSI .....	12
GAMBAR 2.5. PROSES ROTATION/PIVOTING .....	13
GAMBAR 2.6. PROSES SLICE.....	13
GAMBAR 2.7. PROSES DICE DATA.....	14
GAMBAR 2.8. PROSES DRILL DOWN DAN CONSOLIDATION .....	15
GAMBAR 2.9. STAR SCHEMA .....	16
GAMBAR 2.10. SNOWFLAKE SCHEMA .....	17
GAMBAR 2.11. VISUALISASI DIMENSI.....	18
GAMBAR 2.11. VISUALISASI CUBE .....	18
GAMBAR 2.12. HIERARCHY DALAM DIMENSI WAKTU .....	19
GAMBAR 2.13. TABEL FAKTA .....	21
GAMBAR 2.14. PEMBUATAN DIMENSI DENGAN OEM .....	23
GAMBAR 2.15. ARSITEKTUR BI BEANS .....	26
GAMBAR 3.1 USE CASE DIAGRAM.....	28
GAMBAR 3.2. ACTIVITY DIAGRAM – CREATE OLAP METADATA .....	29
GAMBAR 3.3. ACTIVITY DIAGRAM – MODIFY OLAP METADATA .....	29
GAMBAR 3.4. ACTIVITY DIAGRAM – REMOVE OLAP METADATA .....	30
GAMBAR 3.5. ACTIVITY DIAGRAM – VIEW OLAP DATA .....	30
GAMBAR 3.6. ACTIVITY DIAGRAM – CONNECT TO ORACLE OLAP.....	31
GAMBAR 3.7. ACTIVITY DIAGRAM – EXECUTE OLAP PROCESS .....	31
GAMBAR 3.8. ACTIVITY DIAGRAM – PRESENT OLAP DATA.....	32
GAMBAR 3.9. SEQUENCE DIAGRAM – CREATE DIMENSION .....	33
GAMBAR 3.10. SEQUENCE DIAGRAM – CREATE CUBE .....	34
GAMBAR 3.11. SEQUENCE DIAGRAM – MODIFY DIMENSION.....	35
GAMBAR 3.12. SEQUENCE DIAGRAM – REMOVE OLAP METADATA .....	36
GAMBAR 3.13. SEQUENCE DIAGRAM – VIEW DATA .....	36
GAMBAR 3.14. SEQUENCE DIAGRAM – CONNECT TO ORACLE .....	37
GAMBAR 3.15. SEQUENCE DIAGRAM – EXECUTE OLAP PROCESS.....	37
GAMBAR 3.17. CLASS DIAGRAM – OBYEK DIMENSI.....	39
GAMBAR 3.18. CLASS DIAGRAM – OBYEK CUBE .....	41
GAMBAR 3.20. MENU APLIKASI .....	43
GAMBAR 4.1. ARSITEKTUR SISTEM.....	46
GAMBAR 4.6. DIALOG KONFIRMASI KELUAR APLIKASI .....	61
GAMBAR 4.7. SUB MENU OLAP.....	61
GAMBAR 4.8. DIALOG KONFIRMASI MENGHAPUS OBYEK .....	62
GAMBAR 4.9. SUB MENU VIEW .....	62
GAMBAR 4.10. DIALOG ABOUT .....	63
GAMBAR 4.11. CONTEXT MENU UNTUK OBYEK CUBE. ....	64
GAMBAR 4.12. CONTEXT MENU UNTUK OBYEK DIMENSI .....	64
GAMBAR 4.13. TAB SCHEMA PADA FRAMEDIMENSION .....	65



GAMBAR 4.14. TAB TABLES PADA FRAMEDIMENSION .....	66
GAMBAR 4.15. TAB LEVELS PADA FRAMEDIMENSION.....	67
GAMBAR 4.16. TAB HIERARCHIES PADA FRAMEDIMENSION .....	68
GAMBAR 4.17. TAB RELATION PADA FRAMEDIMENSION .....	69
GAMBAR 4.18. TAB DEFINITION PADA FRAMECUBE.....	70
GAMBAR 4.19. TAB FACT TABLE PADA FRAMECUBE .....	71
GAMBAR 4.20. TAB MEASURE PADA FRAMECUBE .....	72
GAMBAR 4.21. TAB DIMENSION PADA FRAMECUBE .....	73
GAMBAR 5.1. DIMENSI STORE .....	78
GAMBAR 5.2. PEMBUATAN DIMENSI STORE .....	80
GAMBAR 5.3. KONFIRMASI SUKSES .....	80
GAMBAR 5.4. DIMENSI TIME.....	81
GAMBAR 5.5. PEMBUATAN DIMENSI TIME.....	83
GAMBAR 5.6. DIMENSI PRODUCT.....	84
GAMBAR 5.7. PEMBUATAN DIMENSI PRODUCT.....	86
GAMBAR 5.8. (A) HASIL PEMBUATAN DIMENSI PADA OEM .....	87
GAMBAR 5.8. (B) HASIL PEMBUATAN DIMENSI PADA BI_CHECKCONFIG .....	87
GAMBAR 5.8. CUBE SALES .....	88
GAMBAR 5.9. PEMBUATAN CUBE SALES.....	89
GAMBAR 5.10. DIALOG KONFIRMASI CUBE .....	90
GAMBAR 5.11. SALES_CUBE PADA OEM .....	90
GAMBAR 5.12. SALES_CUBE PADA BI_CHECKCONFIG .....	91
GAMBAR 5.10. PROSES VIEW CROSSTAB .....	92
GAMBAR 5.11. FRAME VIEW CROSSTAB .....	92
GAMBAR 5.12. DRILL DOWN.....	93
GAMBAR 5.13. ROTATION/PIVOTING .....	94
GAMBAR 5.14. SLICE AND DICE.....	95
GAMBAR 5.15. MODIFY QUERY .....	95
GAMBAR 5.16. CARA MELIHAT GRAFIK .....	96
GAMBAR 5.17. FRAME VIEW GRAPHIC .....	96
GAMBAR B.2. MEMILIH SOURCE DAN DESTINATION .....	104
GAMBAR B.3. INSTALASI BERHASIL.....	105
GAMBAR B.4. CEK KOMPONEN ORACLE OLAP .....	106
GAMBAR C.1. MEMILIH SOURCE DAN DESTINATION .....	107



## DAFTAR TABEL

TABEL 4.2. METODE PADA CLASS OPERATION .....	51
TABEL 4.3. METODE PADA CLASS HIERARCHYDIM.....	53
TABEL 4.4. METODE PADA CLASS LEVELDIM .....	54
TABEL 4.5. METODE PADA CLASS RELATION .....	55
TABLE 5.2. TABEL STORE .....	78
TABEL 5.3. TABEL TIME_BY_DAY.....	81
TABEL 5.4. TABEL PRODUCT_CLASS.....	84
TABEL 5.5. TABEL PRODUCT .....	84
TABEL 5.5. TABEL SALES_FACT .....	88
TABEL C.1. SETING VARIABEL .....	108

TO BELIEVE IS TO FIND THE STRENGTH AND COURAGE THAT  
LIES WITHIN US. WHEN IT IS TIME TO PICK UP THE PIECES  
AND BEGIN AGAIN.

BAB I

PENDAHULUAN



# **BAB I**

## **PENDAHULUAN**

Dalam bab ini akan dijelaskan beberapa hal dasar mengenai tugas akhir ini yang meliputi latar belakang, permasalahan, batasan permasalahan, tujuan, serta sistematika pembahasan tugas akhir ini. Dari uraian tersebut diharapkan, gambaran umum permasalahan dan pemecahan yang diambil dapat dipahami dengan baik.

### **1.1 Latar Belakang**

Kebutuhan sebuah perusahaan terhadap proses analisa terhadap aktivitas bisnisnya, adalah kebutuhan yang tidak bisa ditawar dan ditunda lagi. Perusahaan telah mengembangkan pemikirannya dari era OLTP (On Line Transaction Processing) ke era OLAP (On Line Analytical Processing), yang tidak diragukan lagi menjadi bagian penting bagi pengembangan perusahaan, terutama dalam proses pengambilan keputusan.

Dengan menggunakan teknologi OLAP, pihak penentu kebijakan perusahaan dapat melakukan proses analisis terhadap data-data yang telah mereka miliki sebelumnya. Keunggulan analisis yang dilakukan OLAP adalah analisis data secara multidimensi, artinya mampu menganalisa data dari berbagai sudut pandang. Pada tahap selanjutnya, hasil analisis secara multidimensi ini akan sangat bermanfaat terhadap proses pengambilan keputusan secara cepat dan tepat.

Oracle sebagai DBMS yang banyak digunakan dikalangan korporasi menyediakan fasilitas OLAP yang terintegrasi dalam DBMS-nya. Tidak seperti

versi sebelumnya, pada Oracle9i, Oracle menggabungkan Oracle OLAP sebagai komponen OLAP ke dalam database. Integrasi ini dimaksudkan untuk mendukung proses pengolahan data OLAP sebagai sarana pendukung keputusan dalam proses bisnis perusahaan. Namun, penggunaan teknologi OLAP pada Oracle ini masih dirasa terlalu sulit dan memiliki kelemahan [DAM-03]. Kesulitan terletak pada proses pembentukan metadata OLAP. Kelemahan terletak pada validasi OLAP object (*dimensi*, *cube* dan *measure*) serta prosedur pemanfaatan metadata tersebut.

Untuk itulah, dalam tugas akhir ini akan dibuat sebuah tool untuk memanfaatkan teknologi OLAP pada Oracle secara lebih mudah. Tool ini diharapkan mampu untuk menjawab tantangan penggunaan dan pemanfaatan OLAP sebagai bagian dari sistem data warehouse. Lebih dari itu, tool ini diharapkan mampu memperbaiki kekurangan dan kelemahan dalam sistem yang sudah ada.

Secara spesifik, tool ini diperuntukkan untuk lebih memudahkan dan mempercepat pembentukan metadata OLAP. Selain itu, diharapkan juga dapat menjadi perangkat analisis data multidimensi yang selama ini menjadi keunggulan utama dari teknologi OLAP.

## **1.2 Permasalahan**

Permasalahan yang akan terkait dalam Tugas Akhir ini meliputi beberapa hal sebagai berikut:

1. Bagaimana membuat perangkat lunak yang mampu membentuk metadata OLAP secara mudah dan cepat, namun tetap valid pada database Oracle9i.



2. Bagaimana membuat perangkat lunak yang mampu melakukan modifikasi terhadap metadata OLAP.
3. Bagaimana membuat perangkat lunak yang mampu menampilkan data multidimensional secara mudah untuk keperluan analisis dalam bentuk yang mudah dipahami oleh user.
4. Bagaimana membuat perangkat lunak yang mampu melakukan modifikasi *query* pada data multidimensi dengan mudah tanpa perlu menuliskan perintah dalam bentuk SQL.

### 1.3 Batasan Permasalahan

Dalam pengimplementasian Tugas Akhir ini, dapat disebutkan beberapa batasan yang akan digunakan sebagai acuan, antara lain :

1. Tugas akhir ini tidak membahas bagaimana data ditransformasi dari tabel transaksi ke data warehouse.
2. Database yang dapat dimanfaatkan oleh aplikasi ini adalah Oracle9i Release 2 dengan OLAP option dan mengimplementasikan patch 9.2.0.2.1.
3. OLAP yang akan dibuat berasal dari database relational, bukan dari data *analytic workspace* pada Oracle.
4. Untuk *development* aplikasi OLAP, digunakan Oracle9i Business Intelligence Beans yang terintegrasi dengan Oracle9i JDeveloper 9.0.3.



## **1.4 Tujuan**

Tujuan pembuatan tugas akhir ini adalah merancang dan membuat aplikasi OLAP Builder pada database Oracle9i yang dapat digunakan untuk membangun metadata OLAP serta dapat digunakan untuk pemanfaatan lebih lanjut untuk kepentingan analisis data.

## **1.5 Metodologi Pembuatan Tugas Akhir**

Pembuatan tugas akhir ini terbagi menjadi beberapa tahapan sebagai berikut :

### **a. Studi Literatur**

Mencari dan mempelajari berbagai macam pustaka yang berkaitan dengan rumusan masalah, teori-teori yang berhubungan dengan aplikasi yang akan dibangun, desain sistem, dan bahasa pemrogramannya. Pustaka yang berkaitan dengan tugas akhir ini antara lain Data Warehouse, OLAP, dan Oracle dan bahasa pemrograman Java. Serta pustaka pendukung lainnya mengenai penggunaan OLAP API menggunakan Java DataBase Connectivity (JDBC).

### **b. Desain Perangkat Lunak**

Pada tahap ini akan dilakukan perancangan perangkat lunak, yaitu pemodelan terhadap objek-objek yang terlibat dalam sistem. Pemodelan yang dilakukan meliputi pembuatan use case, sequence diagram dan class diagram.

### **c. Implementasi Perangkat Lunak**

Pada tahap ini akan dilakukan implementasi aplikasi, berdasarkan pemodelan yang sudah dibuat pada tahap sebelumnya. Implementasi aplikasi meliputi perancangan antarmuka aplikasi, pembuatan class-class serta pembuatan fungsi dan procedure yang diperlukan.



d. Uji Coba dan Evaluasi

Pada tahap ini aplikasi akan di uji coba apakah sesuai dengan yang diharapkan. Selanjutnya, akan dilakukan evaluasi terhadap hasil yang diperoleh pada saat uji coba.

e. Penyusunan Buku Tugas Akhir

Pada tahap terakhir ini disusun buku sebagai dokumentasi dari pelaksanaan tugas akhir. Dokumentasi ini juga dibuat sehingga agar pengguna lain dapat memanfaatkan sebagaimana mestinya.

## 1.6 Sistematika Penulisan

Penulisan tugas akhir ini dibagi dalam beberapa bab. Bab I, berisi pendahuluan. Dalam bab ini dibahas latar belakang pembuatan Tugas Akhir, perumusan masalah dan batasan masalah, tujuan serta manfaat yang bisa diperoleh dengan adanya pembuatan, dan metodologi yang digunakan dalam pembuatan Tugas Akhir.

Bab selanjutnya adalah mengenai dasar teori yang digunakan dalam pembuatan Tugas akhir. Bab ini berisi mengenai teori dan konsep pembelajaran yang melandasi pembuatan Tugas Akhir. Dasar teori tersebut meliputi Data Warehouse, On Line Analytical Processing (OLAP) dan Oracle OLAP serta Oracle Business Intelligence Beans (BI Beans).

Bab III membahas mengenai perancangan aplikasi. Bab ini akan menjelaskan proses perancangan perangkat lunak dengan menggunakan bahasa pemodelan, meliputi pembuatan use case diagram, activity diagram, sequence diagram, hingga class diagram.

Bab IV membahas implementasi perangkat lunak berdasarkan desain yang telah dilakukan pada bab sebelumnya. Tahap implementasi meliputi implementasi proses dan antarmuka perangkat lunak.

Bab V membahas uji coba yang dilakukan pada perangkat lunak. Uji coba terdiri dari beberapa skenario untuk menguji fungsionalitas perangkat lunak yang dihasilkan. Pada akhir bab akan disampaikan evaluasi berkaitan dengan kinerja perangkat lunak.

Pada bab VI yang merupakan bab terakhir, diberikan kesimpulan penulis terhadap seluruh proses pembuatan tugas akhir. Saran untuk pengembangan kedepan juga diberikan.



TO BELIEVE IS TO KNOW WE ARE NOT ALONE, THAT LIFE IS A  
GIFT AND THIS IS OUR TIME TO CHERISH IT.

## BAB II

# TEORI PENUNJANG

## **BAB II**

### **TEORI PENUNJANG**

Untuk menunjang konsep perencanaan dan pembuatan Tugas Akhir ini, maka akan dijelaskan beberapa teori dasar mengenai Data Warehouse, Online Analytical Processing, Oracle9i Release 2, serta Oracle9i OLAP sebagai komponen dari Oracle9i R2.

Oracle9i Database Release 2, menyediakan infrastruktur yang terintegrasi untuk implementasi OLAP. Oracle9i database yang didalamnya terdapat komponen Oracle9i OLAP menyediakan kemampuan query dan perhitungan pada database multidimensi.

#### **2.1 Data Warehouse**

Data Warehouse adalah sebuah database relational yang secara spesifik di desain untuk kebutuhan query dan analisis, bukan untuk proses transaksi. Didalamnya mengandung data *history* yang diturunkan dari data transaksi, namun dapat juga berasal dari sumber yang lain.

Data Warehouse bukanlah suatu *Decision Support System* (Sistem Pendukung Keputusan). Tetapi, Data Warehouse adalah sebuah platform yang mengintegrasikan data untuk mendukung aplikasi DSS. Dalam lingkup perusahaan, Data Warehouse meningkatkan produktivitas pengambil keputusan melalui proses konsolidasi.



Data Warehouse merupakan sebuah lingkungan, bukan produk. Data Warehouse adalah sebuah konstruksi arsitektur dari sebuah sistem informasi yang menyediakan informasi kepada user berupa informasi pendukung keputusan dalam klasifikasi waktu. Dengan adanya Data Warehouse suatu perusahaan mampu melakukan proses pengolahan informasi secara efektif, yang pada akhirnya dapat melakukan eksplorasi tren bisnis yang tidak mungkin dilakukan dengan menggunakan sistem biasa.

Untuk menggambarkan Data Warehouse, ada beberapa karakteristik yang perlu dipahami, meliputi :

a. Subject Oriented

Data Warehouse di desain untuk menjawab pertanyaan-pertanyaan spesifik, seperti, “Siapa pelanggan dengan jumlah pembelian terbesar untuk produk A?”, “Apa produk yang paling banyak terjual di negara A selama tahun 1998?”, dan lain-lain.

b. Integrated

Data Warehouse menggabungkan data dari berbagai sumber data transaksi. Sebagai konsekuensinya, didalamnya terdapat aturan-aturan yang memungkinkan data disimpan dalam format yang konsisten.

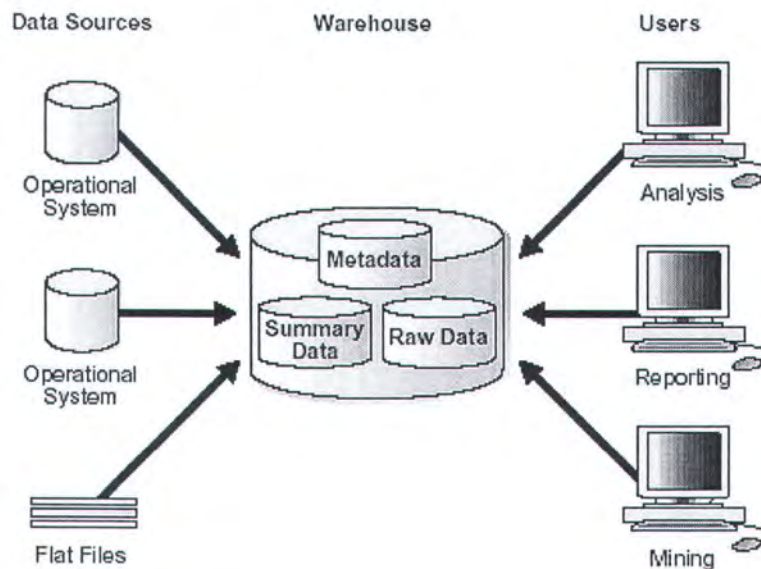
c. Non Volatile

Begitu sekumpulan data dimasukkan ke dalam Data Warehouse, maka data tersebut tidak akan berubah. Hal ini wajar, mengingat tujuan dari Data Warehouse adalah memungkinkan user untuk melakukan proses analisis terhadap data.

d. Time Variant

Untuk bisa menemukan tren dalam bisnis, seorang analis membutuhkan data dalam jumlah besar. Selain itu, data juga merepresentasikan perubahan dari waktu ke waktu. Itulah mengapa Data Warehouse disebut dengan time variant.

Gambar dibawah ini menunjukkan arsitektur dari Data Warehouse.



Gambar 2.1 Arsitektur Data Warehouse

Dalam perspektif bisnis, agar dapat bersaing dalam lingkungan bisnis yang sangat kompetitif, perusahaan memerlukan dukungan sistem yang mampu menjawab tantangan bisnis, karena :

- ✓ Pembuatan keputusan harus dilakukan dengan cepat dan tepat, menggunakan seluruh data yang ada.
- ✓ User merupakan *business domain experts*, bukan profesional dibidang komputer.

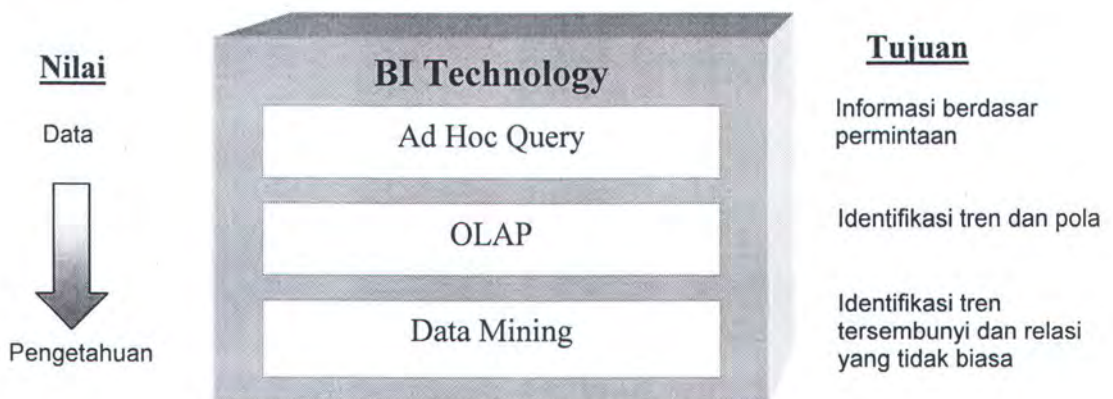


- ✓ Jumlah data pada perusahaan terus bertambah setiap bulan, yang akan berpengaruh pada waktu respon sistem.
- ✓ Persaingan dalam bidang *business intelligence* dan *added information value* semakin meningkat.

## 2.2 On Line Analytical Processing (OLAP)

OLAP adalah suatu metode pemodelan data multidimensional untuk memenuhi kebutuhan analisis terhadap data. OLAP menyajikan data relational kepada user dalam bentuk data multi dimensional untuk memberikan pengetahuan dan pemahaman akan data dan pola-pola tersembunyi.

OLAP memungkinkan analisis pada suatu data yang tidak mungkin dilakukan oleh query biasa. Namun, OLAP juga tidak diperuntukkan untuk melakukan pencarian ilmu pengetahuan terhadap sekumpulan data sebagaimana yang dilakukan dalam Data Mining. Gambar berikut ini menunjukkan posisi teknologi OLAP diantara teknologi-teknologi lain yang memanfaatkan database.



Gambar 2.2 OLAP diantara teknologi lain

Yang membedakan antara OLAP dengan query biasa adalah kedalaman pengolahan data yang dihasilkan. Pada query biasa, informasi yang dihasilkan adalah sesuai dengan permintaan user, sementara pada OLAP informasi direpresentasikan dalam bentuk tren dan pola tertentu yang terdapat pada data. Berbeda dengan keduanya, Data Mining dimaksudkan untuk mengidentifikasi tren tersembunyi dan relasi yang tidak biasa pada data.

2.2.1 Data Multidimensi

Perbedaan mendasar OLAP dengan query biasa adalah kemampuan OLAP dalam menampilkan data dalam format multidimensi. Tabel relasional yang diambil datanya dengan menggunakan *query* hanya mampu menampilkan data dalam bentuk dimensi tunggal. Dengan demikian, hanya ada dua sudut pandang saja, yaitu baris sebagai sumbu x dan kolom sebagai sumbu y. Hal ini ditunjukkan pada gambar 2.3 dimana data dapat dilihat dari dimensi produk yang ada pada baris dan dimensi waktu yang ada pada kolom.

ProductID	1995	1996	...	1997
A001			...	
A002			...	
A003			...	
A004			...	
A005				
A006				

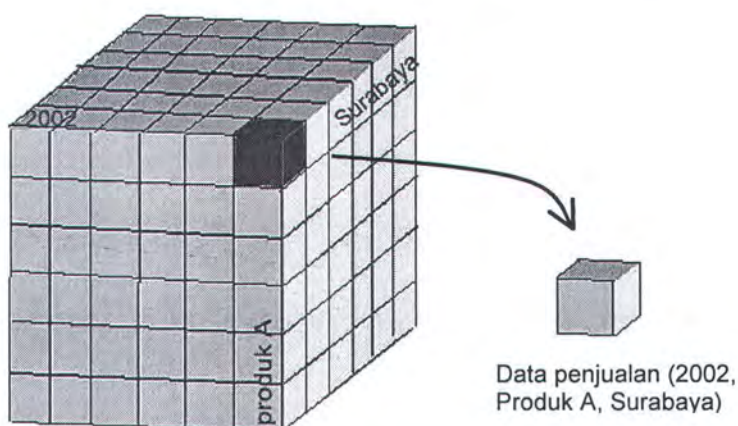
Data satu dimensi, hanya 2 sudut pandang

Gambar 2.3 Data Dimensi Tunggal

Sedangkan yang dimaksud data multidimensi adalah ketika sebuah data dapat dipandang dari berbagai sudut, misal hasil penjualan suatu barang dipandang dari dimensi waktu, lokasi, pembeli dan lain-lain. Kalau digambarkan maka sumbu x



diwakili oleh sudut pandang waktu, sumbu y diwakili oleh produk, dan sumbu z diwakili oleh dimensi lokasi. Hal ini ditunjukkan oleh gambar 2.4.



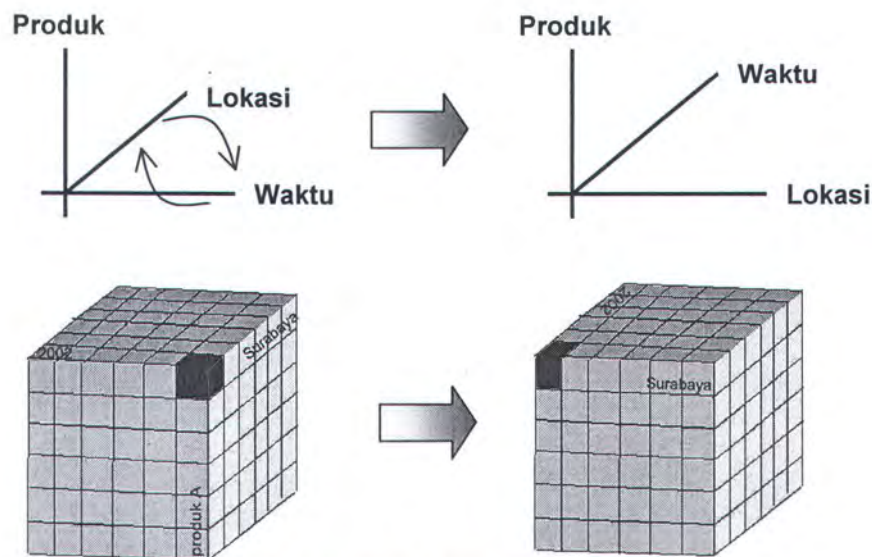
Gambar 2.4 Data multidimensi

## 2.2.2 Operasi Data Multidimensi

Data multidimensi dapat digunakan secara baik untuk keperluan analisis. Untuk itu ada berbagai macam operasi yang dapat dilakukan terhadap data untuk mendapatkan data yang diinginkan.

### 2.2.2.a. Rotation/Pivoting

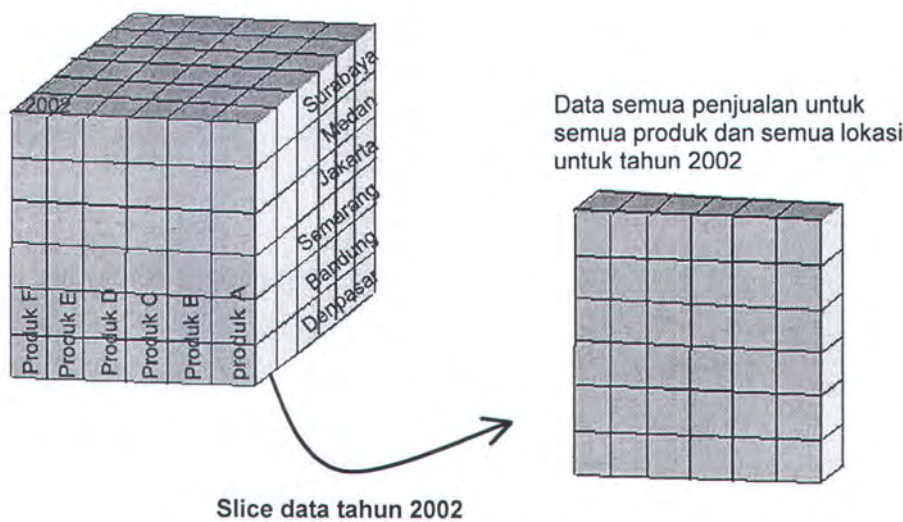
*Rotation/Pivoting* merupakan kemampuan OLAP untuk merubah berbagai macam sudut pandang pada data. Dengan operasi ini, perubahan perspektif sudut pandang menjadi lebih mudah. Proses ini dilakukan dengan melakukan putaran untuk masing-masing dimensi, artinya rotasi suatu sumbu digantikan dengan sumbu yang lain.



Gambar 2.5. Proses Rotation/Pivoting

2.2.2.b Slice dan Dice

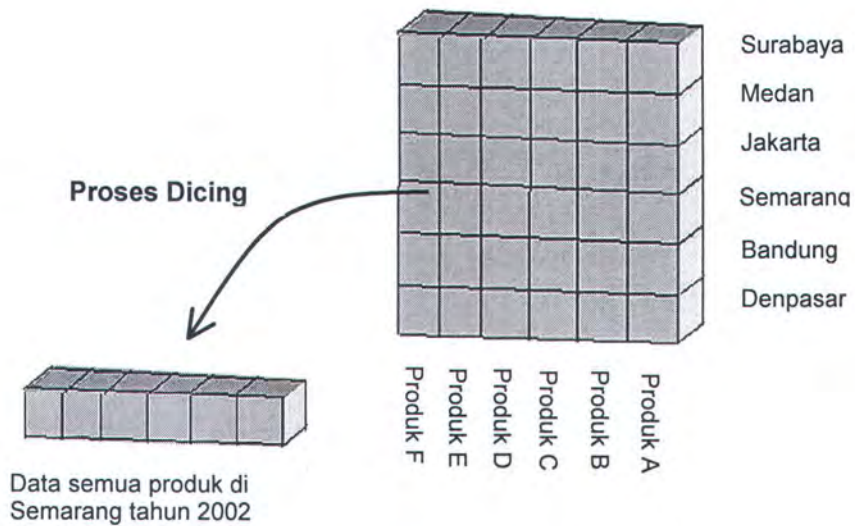
*Slicing* dan *Dicing* merupakan kemampuan OLAP untuk melakukan pemilihan subset pada data. Tiap sumbu pada *cube* diwakili oleh sebuah dimensi. Proses *slice* adalah proses pemotongan data pada *cube* berdasarkan nilai pada satu atau beberapa dimensi.



Gambar 2.6. Proses slice



Sedangkan proses *dice* adalah pemotongan hasil *slice* menjadi bagian subset data yang lebih kecil. Sistem dengan kemampuan *slice* dan *dice* membuat navigasi pada data dengan jumlah sangat banyak menjadi lebih mudah.

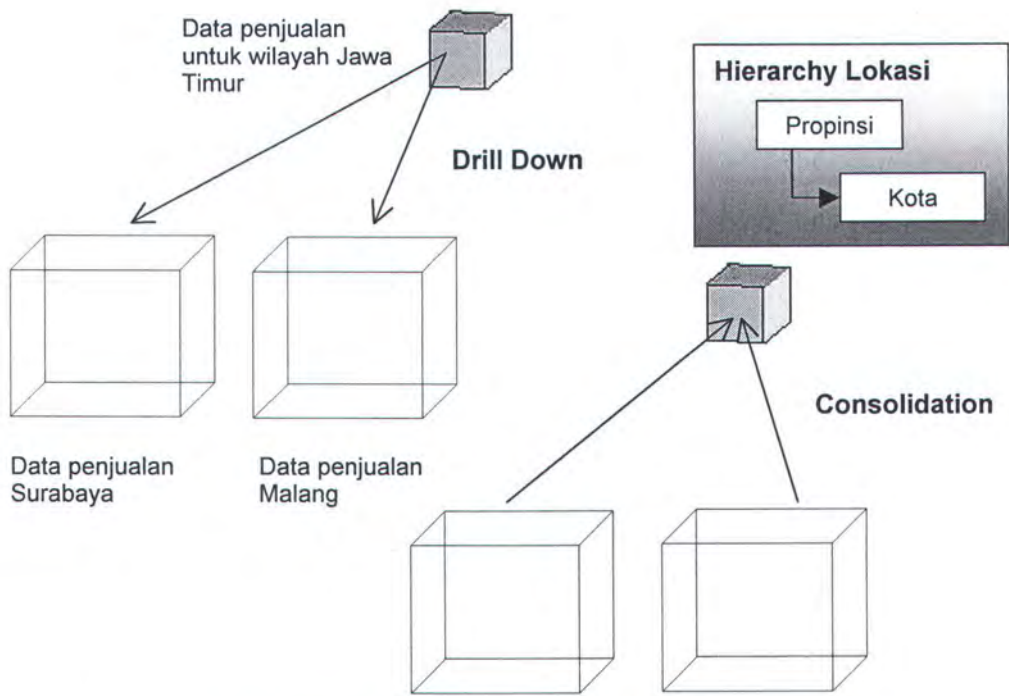


Gambar 2.7. Proses dice data

### 2.2.2.c Drill Down dan Consolidation

Proses *drill down* adalah melihat suatu subset data secara lebih detail. Proses ini melibatkan proses agregasi data. Sebaliknya, *consolidation* melihat data secara global atau rangkuman. Kedua proses ini memanfaatkan hirarki pada dimensi yang membentuk *cube*. Penjelasan lebih lanjut mengenai hirarki akan dibahas pada subbab berikutnya.





Gambar 2.8. Proses drill down dan consolidation

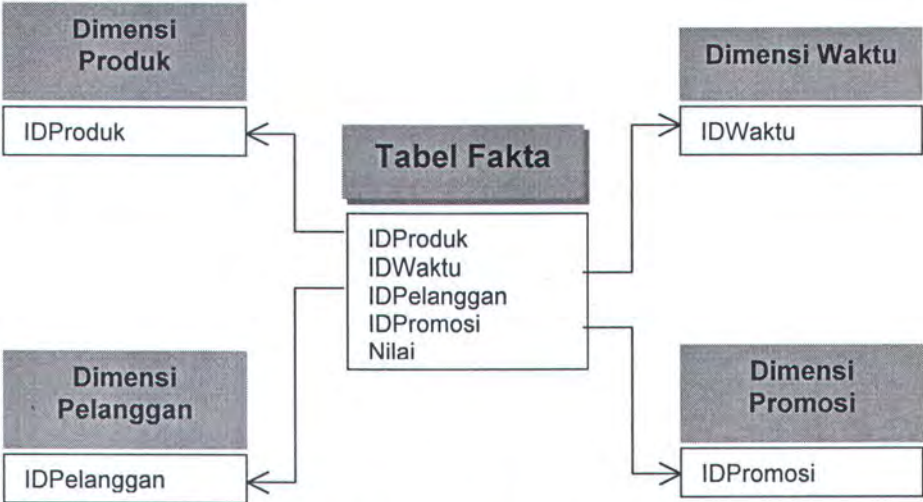
2.2.3 Skema Data Pada OLAP

Skema adalah kumpulan obyek-obyek database. Pada relational data warehouse terdapat dua skema yang digunakan, yaitu star skema dan snowflake skema.

2.2.3.a Star Schema

Skema *star* terdiri dari satu atau lebih tabel fakta dan satu atau lebih tabel dimensi. Tabel fakta merupakan pusat dari *star schema*, karena fungsinya sebagai pengikat dari tabel-tabel dimensi yang terletak disekelilingnya. Hubungan antara tabel-tabel tersebut menggunakan *foreign key*, *metadata* atau keduanya. Gambar dibawah menunjukkan pemodelan database multidimensi dengan *star schema*.



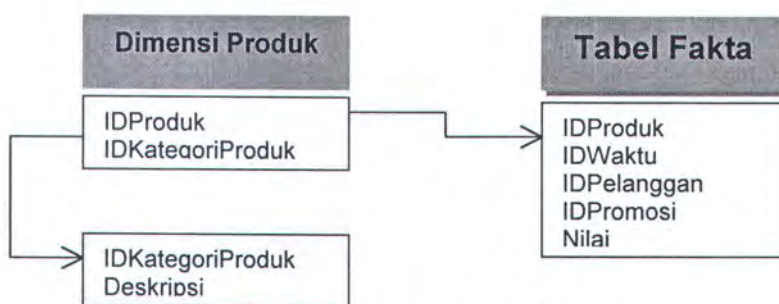


Gambar 2.9. Star Schema

2.2.3.b Snowflake Schema

*Snowflake schema* merupakan pengembangan dari *star schema*. Perbedaannya terletak pada normalisasi data dan jumlah tabel. Pada *snowflake schema* tabel dimensi dinormalisasi secara sebagian atau keseluruhan untuk mengurangi nilai duplikat pada tabel. Hal inilah yang membuat jumlah tabel dimensi pada *snowflake schema* lebih dari satu, tidak seperti pada *star schema*. Sehingga, dengan menggunakan *schema* ini akan diperlukan lebih banyak *join* yang akan membuat *performance* menjadi lebih lambat [ORA-02].

Dalam *snowflake schema*, hanya satu tabel utama yang dihubungkan dengan tabel fakta. Sedangkan tabel-tabel lainnya dihubungkan dengan tabel utama. Gambar dibawah ini menunjukkan pemodelan data menggunakan *snowflake schema*.



Gambar 2.10. Snowflake Schema

## 2.2.4 OLAP Metadata Model

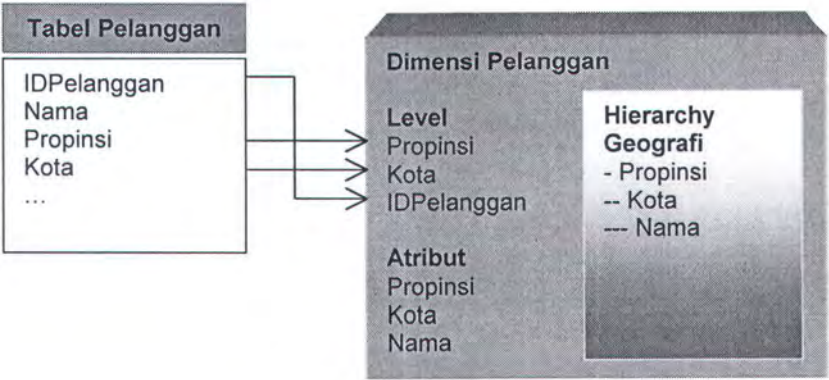
Didalam OLAP *catalog* harus ditentukan apakah sebuah data berperan sebagai dimensi, *attribute*, *measure* atau obyek OLAP yang lain. Hal ini penting dilakukan agar OLAP API dapat mengakses data multidimensi dengan mudah. OLAP memiliki beberapa metadata model yang memiliki fungsi dan peran masing-masing dalam pemrosesan data.

### 2.2.4.a Dimensi

Dimensi adalah sebuah struktur yang terbentuk dari satu atau lebih hirarki yang mengkategorisasi data. Data dimensi dikumpulkan berdasarkan level terendah dari tabel detail, kemudian dilakukan fungsi agregasi ke level-level yang lebih tinggi sehingga lebih berguna untuk proses analisa.

Dimensi terbentuk dari satu atau lebih tabel. Setiap kolomnya merepresentasikan level pada *hierarchy*. Primary key pada tabel pembentuk dimensi akan terhubung dengan tabel fakta. Biasanya, level terendah pada suatu dimensi direpresentasikan dengan kolom yang memiliki nilai unik pada suatu tabel. Gambar dibawah ini menunjukkan obyek dimensi dan obyek-obyek lain yang ada didalamnya.

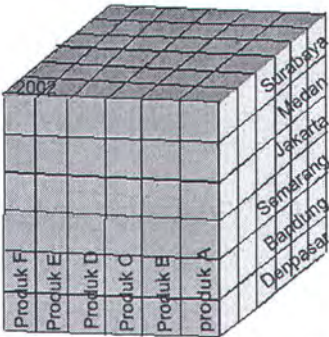




Gambar 2.11. Visualisasi Dimensi

2.2.4.b Cube

Cube adalah obyek OLAP yang tersusun dari *measure*, *dimension* dan *attribute* [ORA-02]. Sisi-sisi pada cube ditentukan oleh masing-masing dimensi yang terlibat dalam *cube* itu. Seperti pada gambar dibawah ini, sisi *cube* menggambarkan tiga dimensi yang terlibat yaitu, dimensi waktu, dimensi lokasi dan dimensi produk. Meskipun tidak ada batasan jumlah dimensi pada cube, namun biasanya untuk alasan kemudahan tampilan digunakan tiga dimensi yang mewakili sisi baris, sisi kolom dan sisi *page*.



Gambar 2.11. Visualisasi cube

#### 2.2.4.c Hierarchy

*Hierarchy* adalah salah satu OLAP objek yang menggunakan level secara bertingkat untuk mengorganisasi data. Sebuah hirarki dapat digunakan untuk mendefinisikan agregasi data. Sebagai contoh pada dimensi waktu, hirarki mungkin akan melakukan agregasi data dari level bulan ke level tiga-bulanan selanjutnya ke level tahun. Dalam *hierarchy*, sebuah level dihubungkan dengan level di atasnya dan dibawahnya. Sebuah dimensi dapat terdiri dari lebih dari satu *hierarchy*, misalnya dimensi produk mungkin terdiri dari *hierarchy* kategori produk dan *hierarchy* suplier produk.



Gambar 2.12. Hierarchy dalam dimensi waktu

#### 2.2.4.d Level

Level merepresentasikan sebuah posisi pada hirarki. Level mengumpulkan data untuk agregasi dan digunakan untuk proses komputasi. Sebagai contoh, pada dimensi waktu, level yang mungkin didefinisikan adalah level hari, level bulan, level triwulan, dan level tahun. Setiap level diatas level terendah merupakan agregasi dari level dibawahnya. Jika data penjualan disimpan dalam format harian



(level hari), maka data dapat diagregasi sesuai dengan level yang ada di atasnya, seperti level bulan, dan tahun.

#### 2.2.4.e Attribute

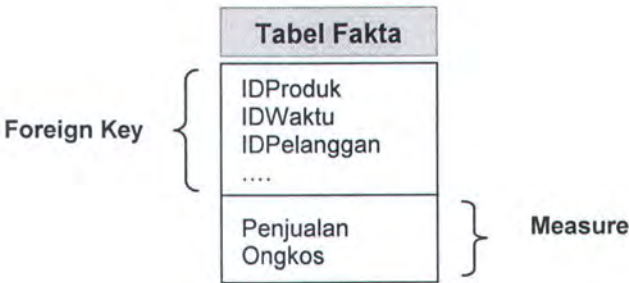
*Attribute* merepresentasikan informasi tambahan pada sebuah level tertentu. Sebuah level dapat memiliki lebih dari satu atribut, tetapi minimal harus memiliki satu atribut. Nilai atribut berguna sebagai nilai yang akan mewakili level ketika data multidimensi ditampilkan kepada user. Hal ini disebabkan tidak semua nilai pada level bisa dimengerti dan dipahami oleh user. Misalnya, level *product name* menyimpan nilai *product id*, sedangkan atributnya menyimpan nilai dari *product brand*, dengan demikian yang akan ditampilkan kepada user ketika user memilih level *product name* adalah nilai pada kolom *product brand*, bukan *product id*.

#### 2.2.4.f Tabel Fakta

Tabel fakta merupakan pusat dari skema pada OLAP. Tabel fakta mempunyai dua tipe kolom, yaitu kolom yang menyimpan nilai-nilai numerik (disebut dengan *measure*) dan kolom yang menyimpan *foreign key* yang mengacu ke tabel dimensi, seperti yang ditunjukkan pada gambar 2.13. Nilai numerik yang ada pada tabel fakta merupakan nilai agregat dari data-data yang berasal dari tabel dimensi. Hubungan antara tabel fakta dengan tabel dimensi adalah *one to many*, sehingga masing-masing *primary key* dari tabel dimensi dijadikan key acuan pada tabel fakta. Dengan demikian, tabel fakta menyimpan setiap kombinasi *key* tabel dimensi yang melingkupinya.

2.2.4.g Measure

*Measure* dapat diartikan sebagai fakta. Nilai *measure* terletak pada tabel fakta. *Measure* mengandung data yang akan dianalisa seperti penjualan ongkos produksi. OLAP *catalog* memerlukan informasi kolom bertipe numeric yang akan dijadikan *measure*. Seringkali, sebuah *measure* memiliki data dengan tipe numeric agar dapat dilakukan proses penambahan terhadapnya [ORA-02].



Gambar 2.13. Tabel Fakta

2.3 Oracle9i OLAP

Pada versi sebelumnya, Oracle memisahkan OLAP Service dengan Oracle database. Namun pada Oracle9i, OLAP Service menjadi bagian integral dari Oracle database. Dengan digabungkannya komponen OLAP ke dalam database akan memudahkan pemakai database untuk melakukan operasi-operasi analisis terhadap data.

Oracle OLAP memungkinkan user untuk menjalankan fungsi-fungsi analisis secara langsung dari dalam database, sehingga kinerja database dapat ditingkatkan secara signifikan.



### 2.3.1 Oracle9i OLAP API

Oracle OLAP API adalah bahasa query yang dapat digunakan untuk memilih dan memanipulasi data-data analisa untuk ditampilkan kepada user. OLAP API melakukan koneksi dengan database menggunakan JDBC. Lebih jauh lagi, OLAP API adalah teknologi yang digunakan oleh Oracle BI Beans untuk mengakses relational dan multidimensional data.

### 2.3.2 OLAP Catalog

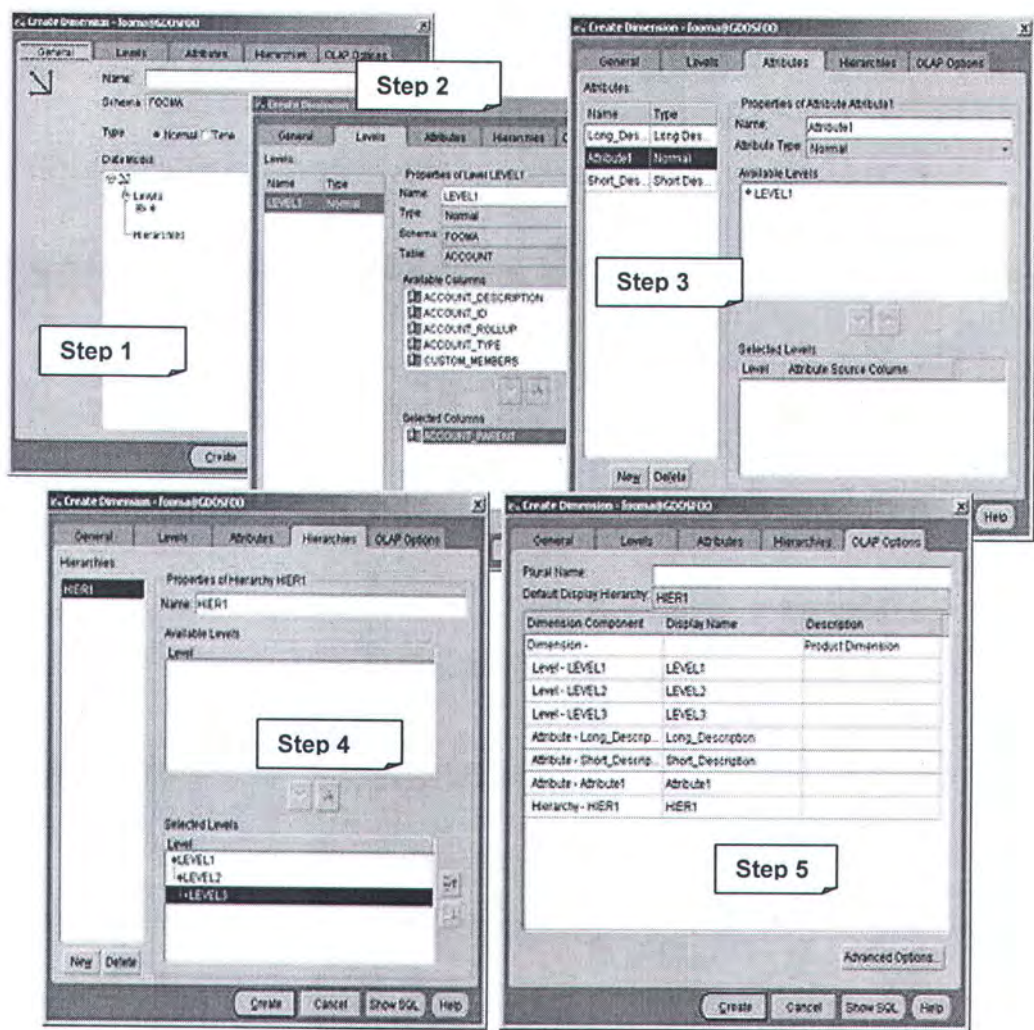
OLAP Catalog Metadata dibuat dan disimpan pada tabel relasional di dalam database. Aplikasi OLAP dapat mengakses OLAP Catalog ini untuk mengetahui data yang tersedia, dianalisa kemudian ditampilkan. Metadata berisi informasi mengenai lokasi fisik data, apakah disimpan di dalam tabel relasional atau *analytic workspace*.

Dimanapun lokasi penyimpanan, tabel relasional atau *analytic workspace*, data multidimensional disimpan dalam bentuk multidimensional objek, yaitu measures, dimension, level dan attribute. Metadata menyimpan informasi yang berhubungan dengan pemilihan, manipulasi dan presentasi data.

### 2.3.3 Membuat OLAP Metadata Dengan OEM

Untuk membuat metadata OLAP, diperlukan pengetahuan tentang data pada tabel yang akan dijadikan dimensi. Oracle Enterprise Manager (OEM) menyediakan fasilitas untuk membuat dimensi berdasarkan data pada suatu tabel. Beberapa hal yang dibutuhkan sebagai informasi untuk membuat dimensi dengan OEM adalah :

- 1. Nama dimensi
- 2. Tabel data untuk dimensi
- 3. Nama tiap level dan kolom yang mengacunya.
- 4. Jumlah dan susunan level pada tiap *hierarchy* yang dibuat.
- 5. Join key yang menghubungkan level jika dimensi lebih dari satu tabel.
- 6. Kolom yang mengandung atribut untuk tiap level
- 7. Deskripsi dan *display name* dari dimensi, *hierarchy*, level dan *attribute*.



Gambar 2.14. Pembuatan dimensi dengan OEM



Pada gambar 2.14 dapat dilihat proses pembuatan dimensi dengan model *star schema*. Proses dibagi menjadi lima langkah yang direpresentasikan dengan lima *tab componen*. Langkah 1, mendefinisikan nama dimensi yang akan dibuat. Langkah 2, mendefinisikan level dan kolomnya. Langkah 3, mendefinisikan *attribute* dan kolom untuk setiap level yang ada. Langkah 4, mendefinisikan *hierarchy* dan susunan level yang membentuknya. Dan langkah 5, menentukan deskripsi untuk masing-masing obyek yang didefinisikan sebelumnya.

Selain menggunakan OEM, pembuatan dimensi juga dapat dilakukan dengan menggunakan PL/SQL. Hanya, untuk menggunakan bahasa PL/SQL dibutuhkan pengetahuan tentang *package* pada Oracle yang berguna untuk proses pembuatan dimensi.

OEM juga menyediakan fasilitas untuk mendefinisikan metadata OLAP yang lain seperti *cube* dan *measures*.

## 2.4 Oracle JDBC

Oracle Java DataBase Connectivity (JDBC) merupakan sebuah library pada Oracle yang memungkinkan para programmer melakukan koneksi dari sebuah aplikasi (Java Client) ke database Oracle. Oracle JDBC mengimplementasikan *interface* paket `java.sql` dari Sun Microsystem. Oracle menawarkan 3 JDBC driver standar sebagai berikut :

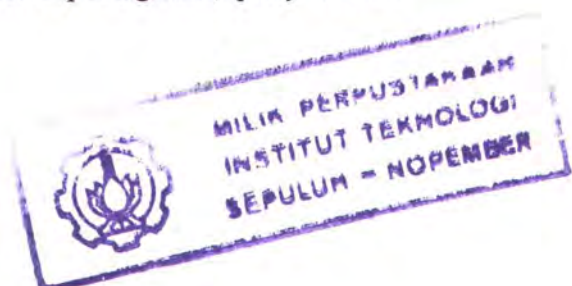
1. JDBC Thin Driver, dapat digunakan pada aplikasi di sisi client atau applet dan tidak memerlukan instalasi Oracle client.

2. JDBC OCI Driver, digunakan untuk aplikasi client namun memerlukan proses instalasi Oracle client.
3. Server-side JDBC Driver, dimasukkan kedalam Oracle9i Server.

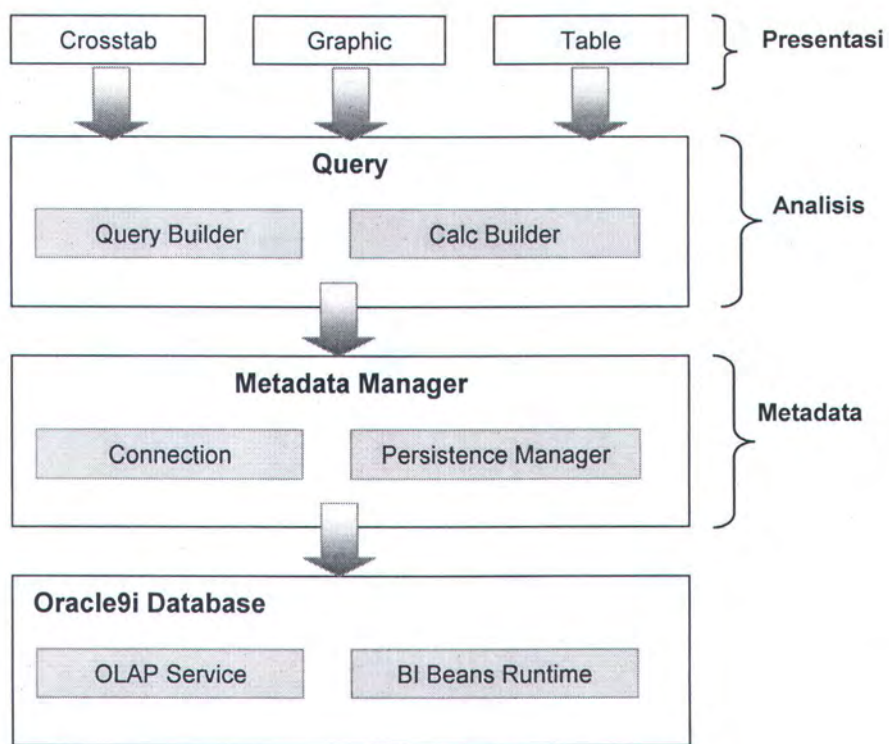
## 2.5 Oracle Business Intelligence Beans

Oracle9i Business Intelligence Beans (BI Beans), menyediakan komponen aplikasi yang *reusable* yang dapat digunakan untuk pengembangan aplikasi *business intelligence* secara cepat. Untuk penyajian data dan analisa dalam bentuk aplikasi OLAP, digunakan komponen Java BI Beans. Objective dari BI Beans adalah untuk menyediakan komponen aplikasi yang reusable yang akan mempercepat proses pengembangan aplikasi.

Oracle9i BI Beans merupakan sekumpulan standar javabean yang menyediakan fitur-fitur dalam membangun aplikasi analisa pada Oracle9i OLAP. Oracle9i BI Beans terbagi menjadi tiga kategori yaitu *presentation*, *OLAP* dan *catalog service (BI Beans catalogs)*. *Presentation beans* digunakan untuk penyajian data, yang didalamnya terdapat grafik dengan berbagai macam tipe, *crosstab* dan tabel yang akan diperlihatkan pada user. OLAP Beans akan mempermudah dalam membuat suatu perintah OLAP SQL yang jika ditulis secara biasa akan memakan ratusan baris sql, namun dengan OLAP Beans dengan fasilitas Query Buildernya mampu membuat hanya dalam waktu yang relatif lebih singkat. BI Beans catalog digunakan untuk menyimpan, menerima dan mengelola semua obyek yang dibuat developer seperti report, grafik, query favorit.







Gambar 2.15. Arsitektur BI Beans

TO BELIEVE IS TO KNOW THAT EVERY DAY IS A NEW  
BEGINNING. IT IS TO TRUST THAT MIRACLES HAPPEN, AND  
DREAMS REALLY DO COME TRUE.

### BAB III

## DESAIN PERANGKAT LUNAK

## BAB III

### DESAIN PERANGKAT LUNAK

Dalam bab ini akan dijelaskan mengenai tahapan desain perangkat lunak yang akan dibagi menjadi tiga bagian yaitu, desain proses, desain obyek dan desain antarmuka.

#### 3.1 Desain Proses

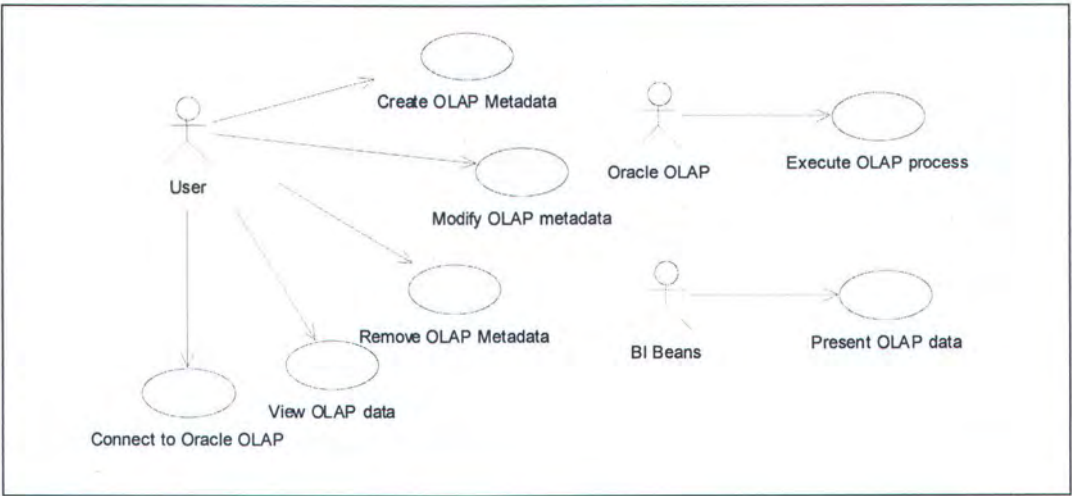
Untuk menjelaskan proses yang ada pada perangkat lunak ini akan digunakan penggambaran model menggunakan konsep *Unified Modelling Language* (UML).

##### 3.1.1 Use Case Diagram

Dalam *use case diagram* terdapat tiga *actor* yang terlibat dalam perangkat lunak yaitu, User, Oracle OLAP dan Oracle BI Beans. Aktor User adalah pengguna perangkat lunak yang dapat melakukan koneksi Oracle OLAP, membuat OLAP metadata, *modify* OLAP metadata serta melihat data multidimensi OLAP. Aktor Oracle OLAP adalah server Oracle yang akan melakukan semua proses eksekusi OLAP. Sedangkan Oracle BI Beans berperan untuk menampilkan data multidimensi.

Berikut adalah *use case diagram* untuk perangkat lunak ini :





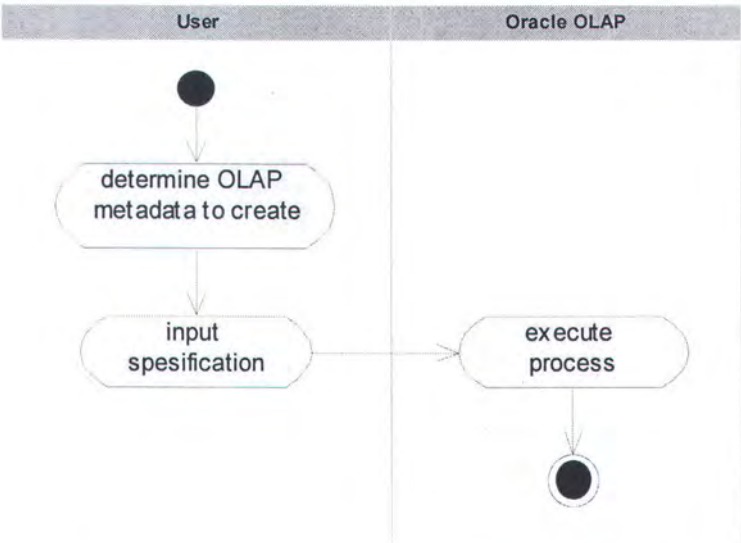
Gambar 3.1 Use Case Diagram

3.1.2 Activity Diagram

Berikut ini beberapa *activity diagram* untuk menjelaskan *use case diagram* di atas.

- **Create OLAP metadata**

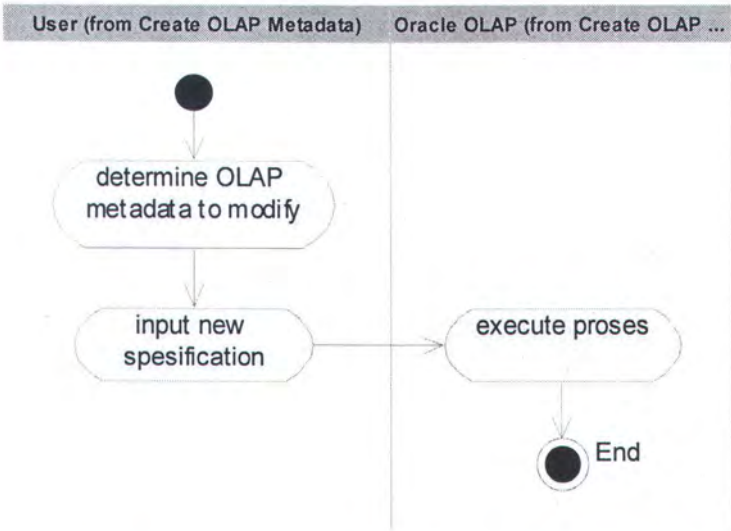
User menentukan metadata OLAP yang akan dibuat, kemudian menentukan spesifikasi obyek. Selanjutnya Oracle OLAP akan mengeksekusi proses sesuai dengan spesifikasi user.



Gambar 3.2. Activity Diagram – Create OLAP Metadata

- **Modify OLAP Object**

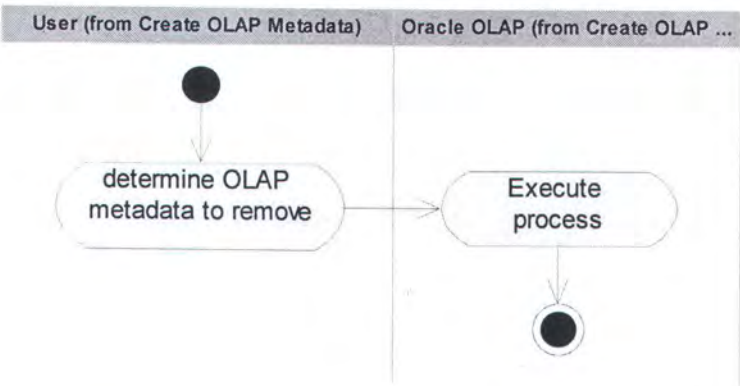
User menentukan perubahan pada metadata OLAP yang sudah ada, selanjutnya akan dieksekusi oleh Oracle OLAP.



Gambar 3.3. Activity Diagram – Modify OLAP metadata

- **Remove OLAP Metadata**

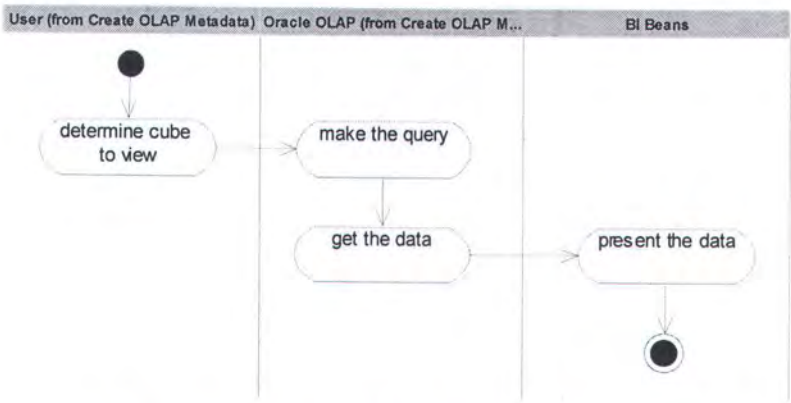
User menentukan obyek OLAP yang akan dihapus, kemudian akan dieksekusi oleh Oracle OLAP.



Gambar 3.4. Activity Diagram – Remove OLAP Metadata

- **View OLAP Data**

Untuk mendapatkan query yang akan dieksekusi oleh Oracle OLAP, user harus menentukan *cube* terlebih dahulu, dan hasilnya akan dikirimkan ke BI Beans untuk ditampilkan dalam format yang diinginkan.

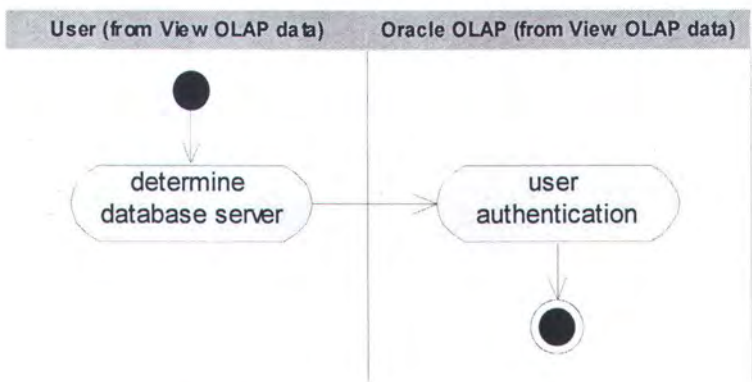


Gambar 3.5. Activity Diagram – View OLAP Data



- **Connect to Oracle OLAP**

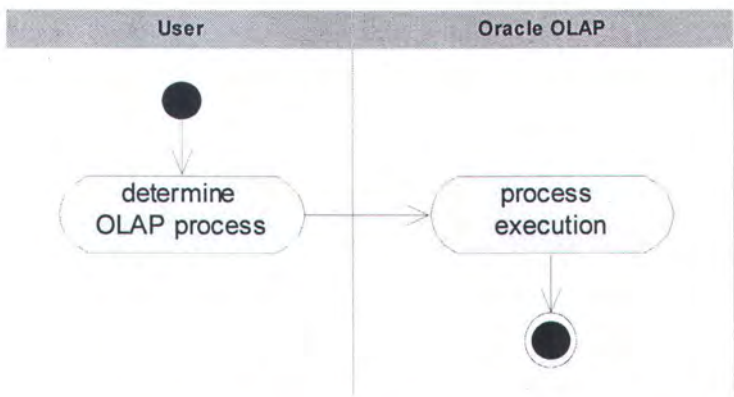
Untuk melakukan koneksi ke Oracle OLAP, user harus mengisi informasi berupa username, password dan service.



Gambar 3.6. Activity Diagram – Connect to Oracle OLAP

- **Execute OLAP Process**

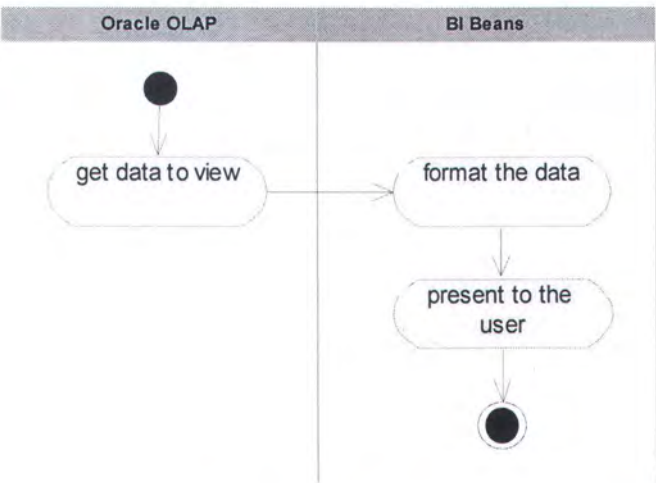
Tipe dan jenis OLAP proses yang akan dieksekusi oleh Oracle OLAP ditentukan oleh user dengan cara memilih jenis pekerjaan yang ingin dilakukan.



Gambar 3.7. Activity Diagram – Execute OLAP Process

- **Present OLAP Data**

Data hasil query yang dilakukan oleh Oracle OLAP akan dikirimkan ke BI Beans untuk diformat dan dipresentasikan kepada user.



Gambar 3.8. Activity Diagram – Present OLAP Data

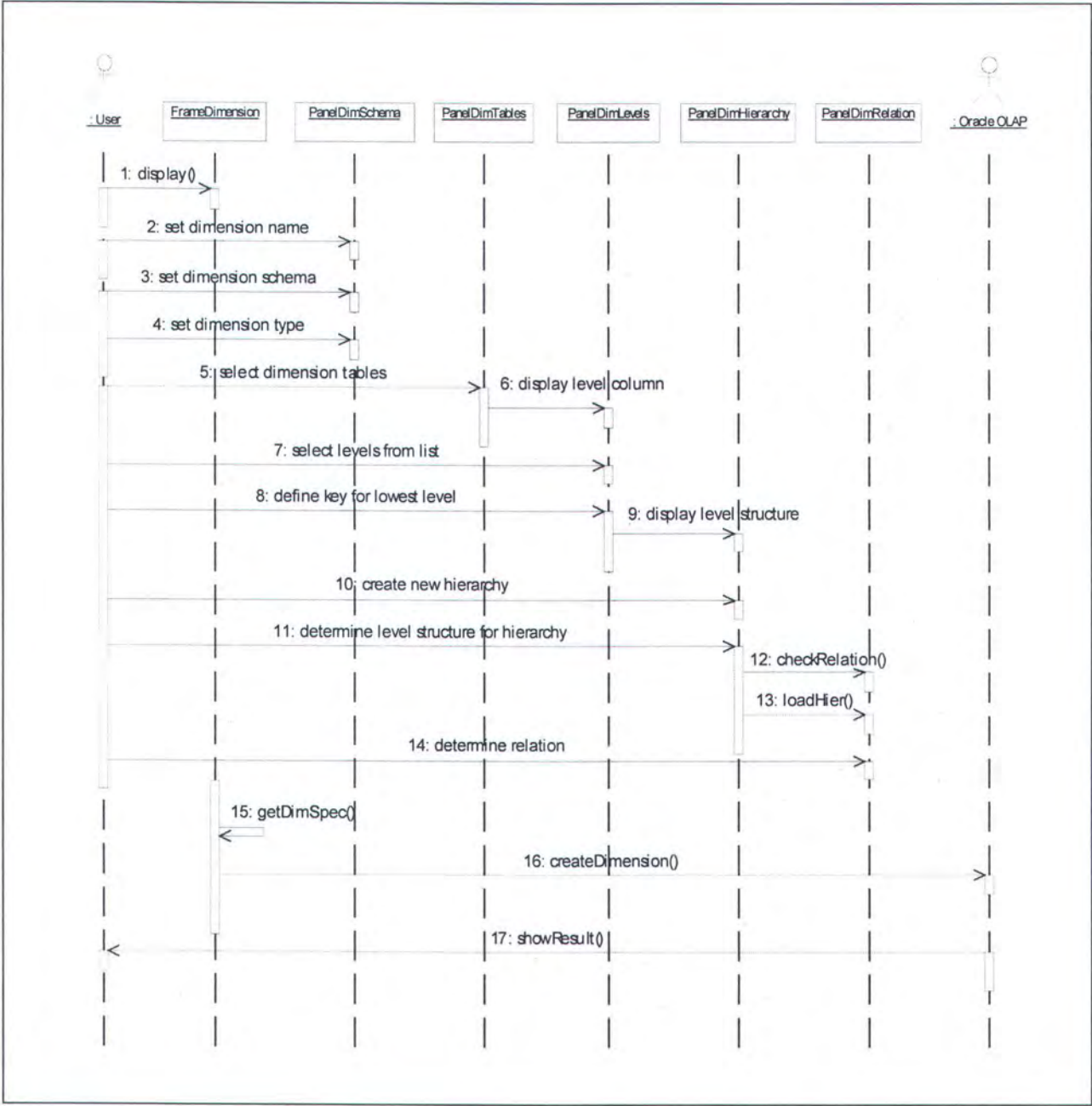
**3.1.3 Sequence Diagram**

Berikut ini adalah *sequence diagram* untuk kedelapan *use case* di atas.

- **Create OLAP Metadata**

*Use case* ini memiliki dua *sequence diagram*, yaitu Create dimension dan Create Cube.



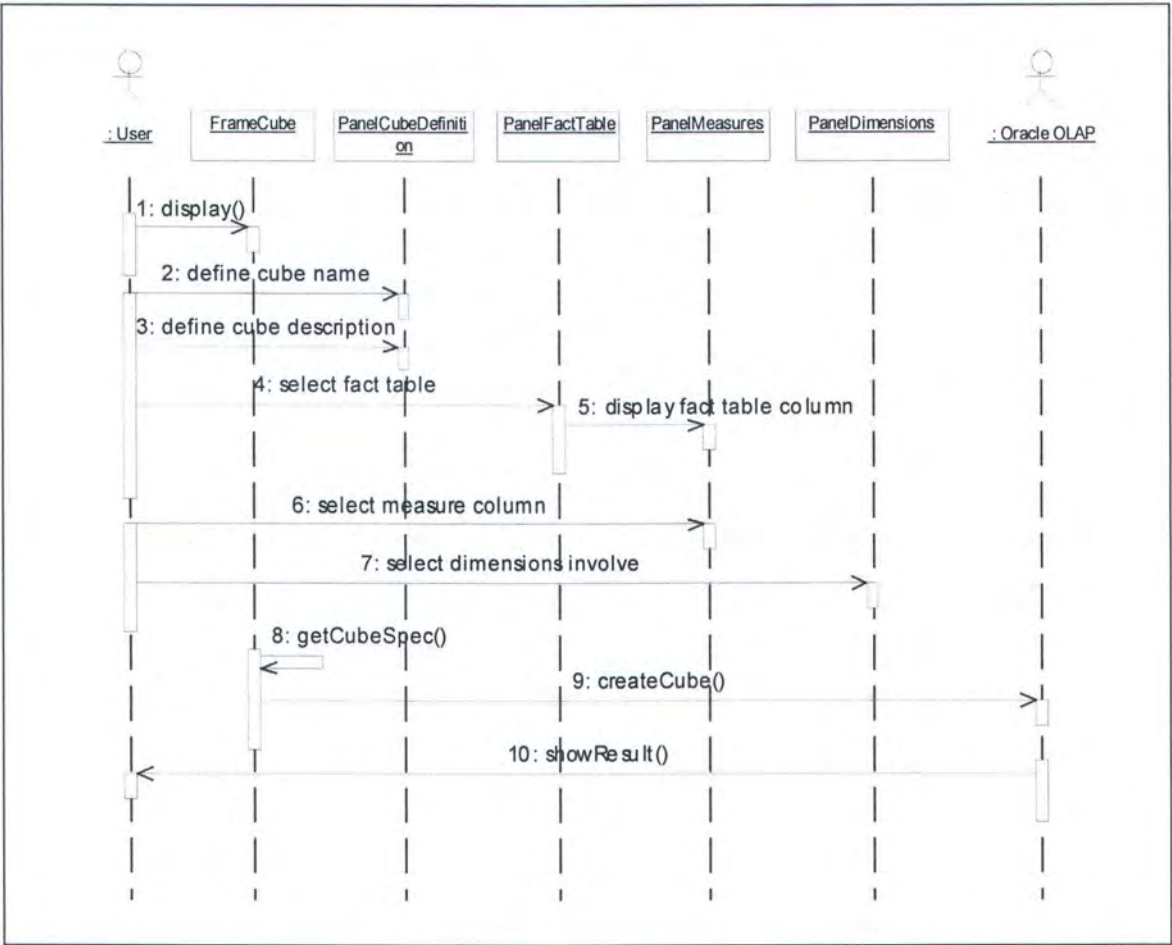


Gambar 3.9.Sequence Diagram – Create Dimension

User menginputkan informasi tentang spesifikasi dimensi yang akan dibuat. spesifikasi meliputi nama dimensi, skema dimensi, tipe dimensi, tabel dimensi, level-level yang terlibat, *hierarchy* serta struktur level yang membentuknya.

Spesifikasi dari user tersebut selanjutnya akan dieksekusi oleh Oracle OLAP, dan akan mengirimkan hasilnya.

Untuk membuat *cube*, user menginputkan sejumlah informasi yang merupakan spesifikasi *cube* seperti nama *cube*, deskripsi *cube*, tabel fakta, dimensi serta *measure*. Selanjutnya Oracle OLAP akan membuat *cube* berdasarkan spesifikasi tersebut.

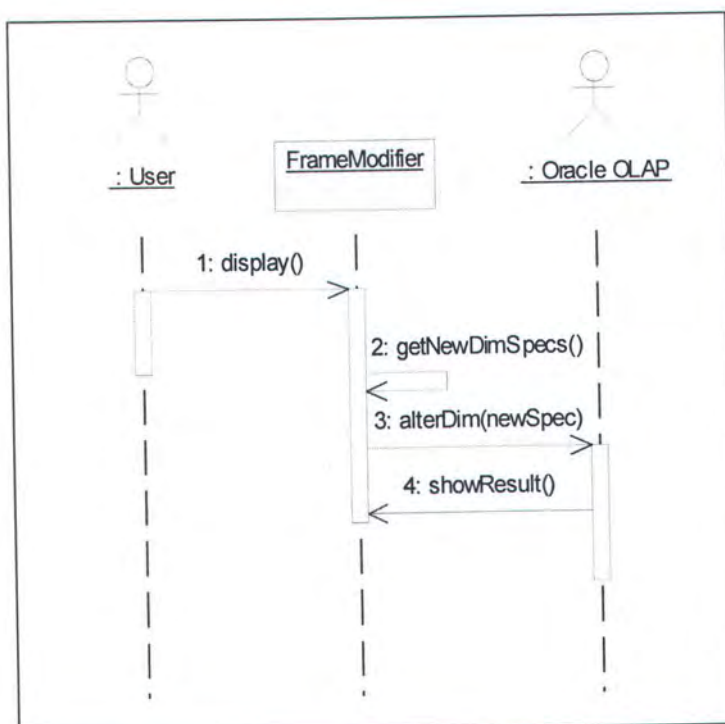


Gambar 3.10. Sequence Diagram – Create Cube



- **Modify OLAP Metadata**

Aktivitas modifikasi metadata hanya dilakukan untuk metadata dimensi saja. Sehingga hanya dibutuhkan sebuah *sequence diagram* seperti ditunjukkan gambar berikut.



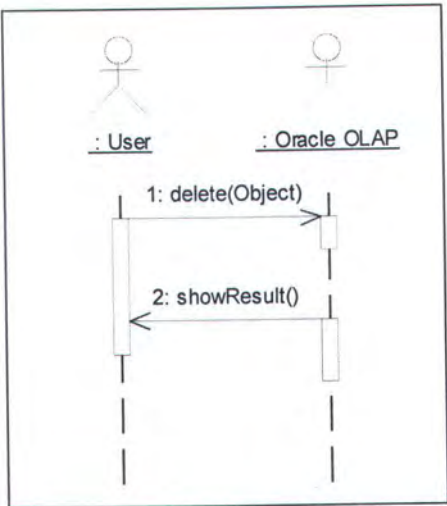
Gambar 3.11. Sequence Diagram – Modify Dimension

Mengedit dimension dilakukan dengan menginputkan perubahan spesifikasi pada dimensi yang dimaksud. Perubahan ini selanjutnya akan di-*alter* oleh Oracle OLAP.

- **Remove OLAP Metadata**

User mengirimkan informasi obyek yang akan dihapus kepada Oracle OLAP. Selanjutnya Oracle OLAP akan mengeksekusi perintah dan memberitahukan hasilnya kepada user.

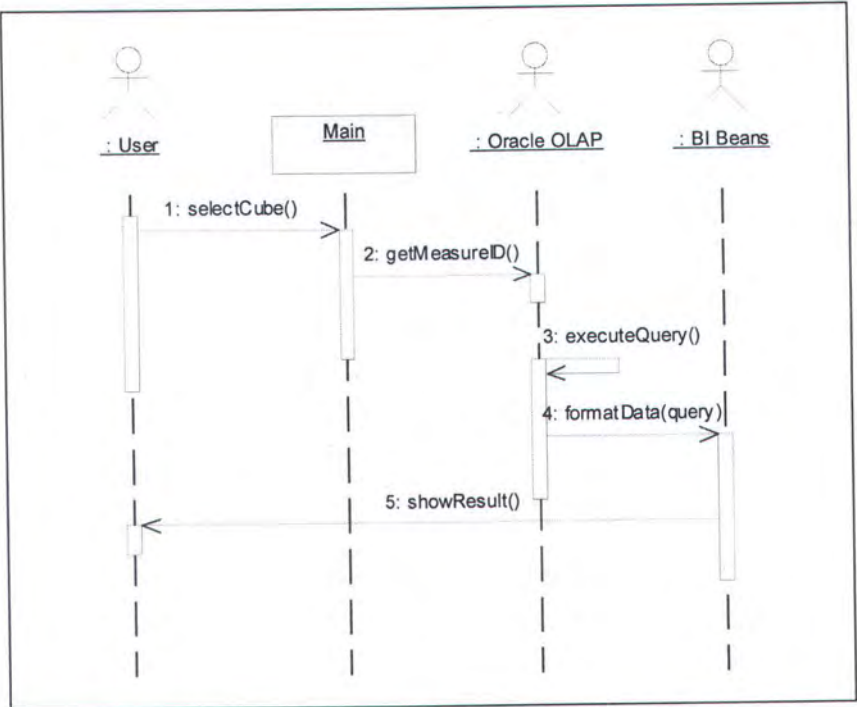




Gambar 3.12. Sequence Diagram – Remove OLAP Metadata

- **View OLAP Data**

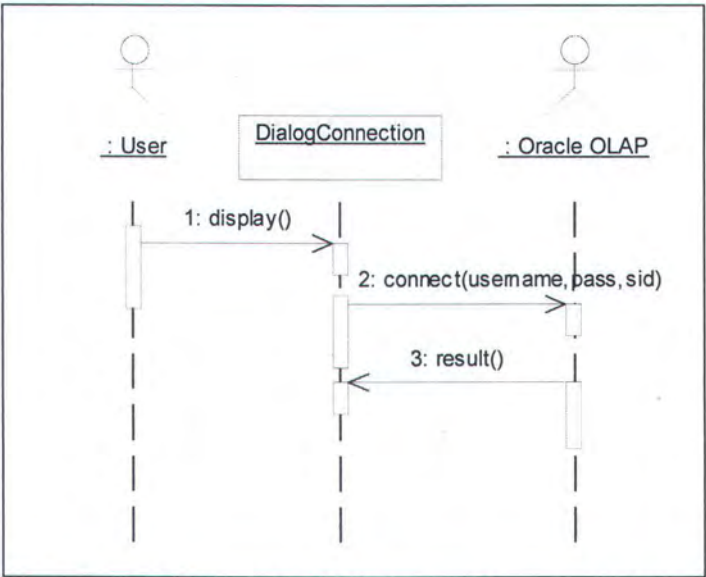
Untuk melihat data OLAP, user menentukan *cube* yang akan dilihat datanya. Kemudian Oracle OLAP akan mencari `measureID` dan menentukan querynya. *Resultset* data kemudian akan tampilkan oleh BI Beans.



Gambar 3.13. Sequence Diagram – View Data

- **Connect to Oracle OLAP**

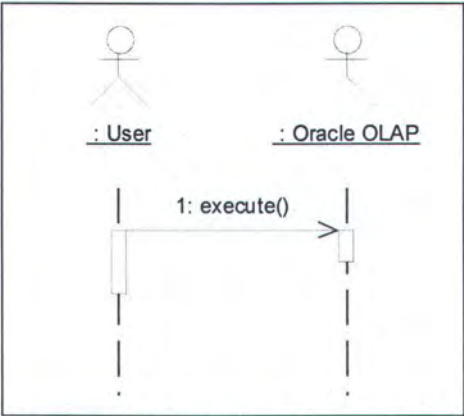
Koneksi ke Oracle OLAP dilakukan dengan mengisi username, password dan service database pada *dialog* Koneksi.



Gambar 3.14. Sequence Diagram – Connect to Oracle

- **Execute OLAP Process**

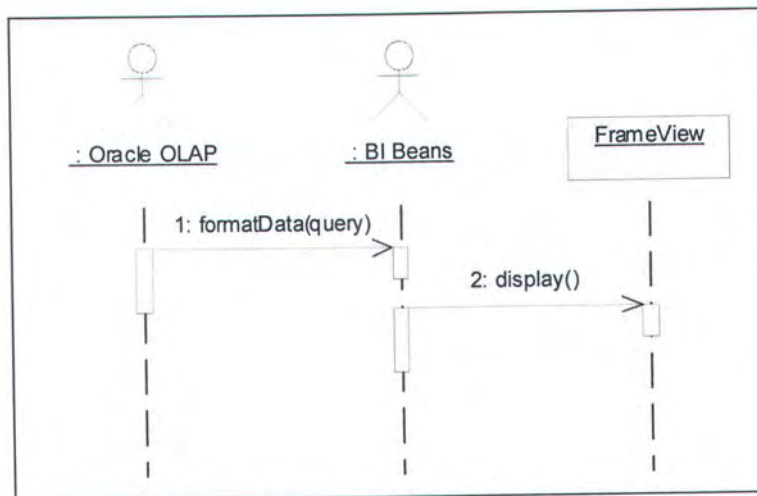
Mengeksekusi proses-proses OLAP dilakukan oleh Oracle



Gambar 3.15. Sequence Diagram – Execute OLAP Process

- **Present OLAP Data**

BI Beans menerima hasil query dari Oracle OLAP yang selanjutnya akan ditampilkan pada window *FrameView*.



Gambar 3.16. Sequence Diagram – Present Data

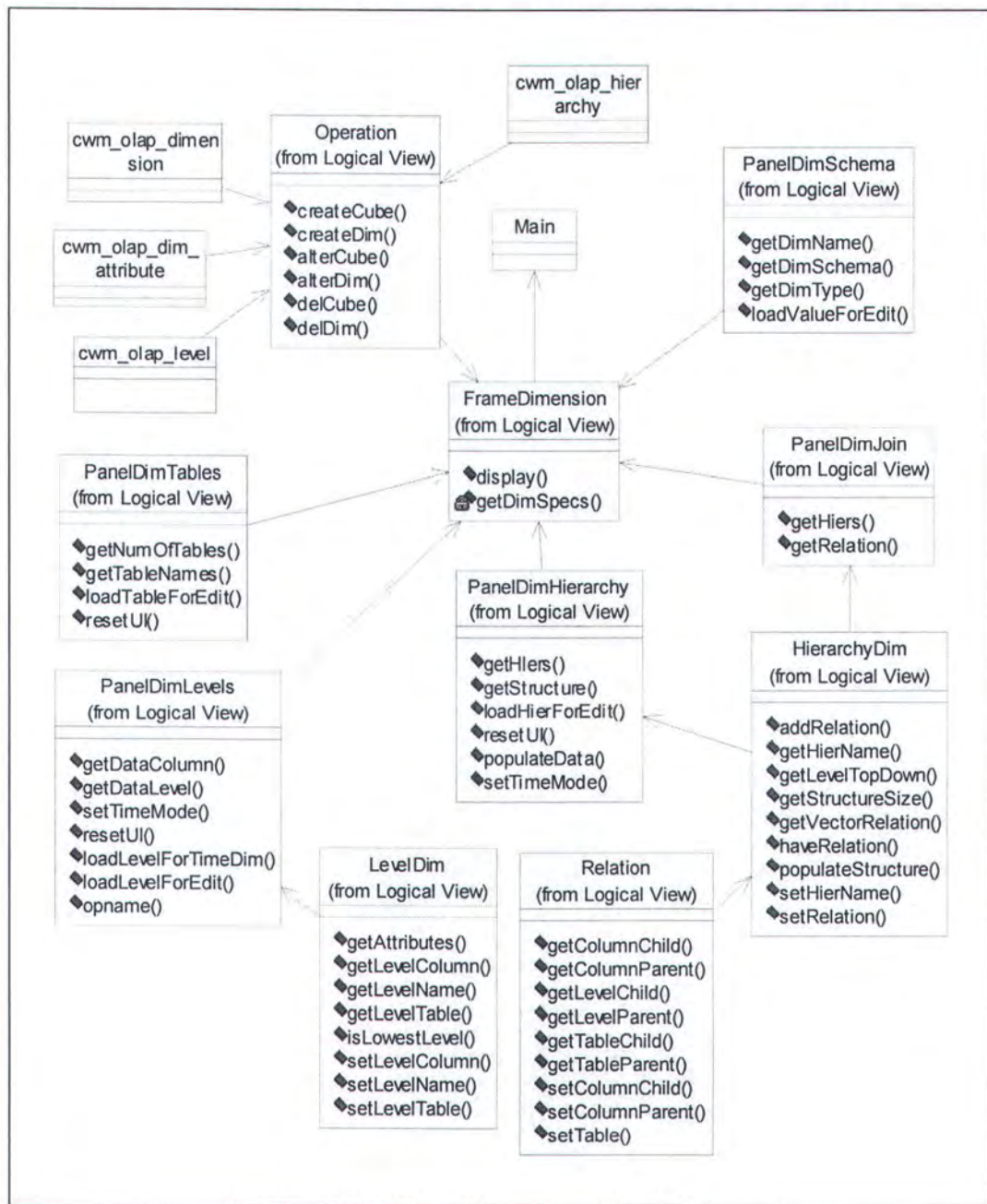
### 3.2 Desain Obyek

Pembuatan obyek terdiri dari 3 bagian yang menunjukkan masing-masing proses, yaitu proses dimensi, proses cube dan proses presentasi data. Masing-masing proses ini mewakili aktivitas yang bisa dilakukan oleh user dalam perangkat lunak ini.



### 3.2.1 Obyek Pada Proses Dimensi

Proses-proses yang berhubungan dengan dimensi, dilakukan dengan menggunakan *class* *FrameDimension*. *Class* ini merupakan turunan dari *class* *JFrame* pada Java. *FrameDimension* menggabungkan *class* *PanelDimTables*, *PanelDimLevels*, *PanelDimHier* dan *PanelDimJoin* dalam sebuah tampilan.



Gambar 3.17. Class Diagram – Obyek Dimensi

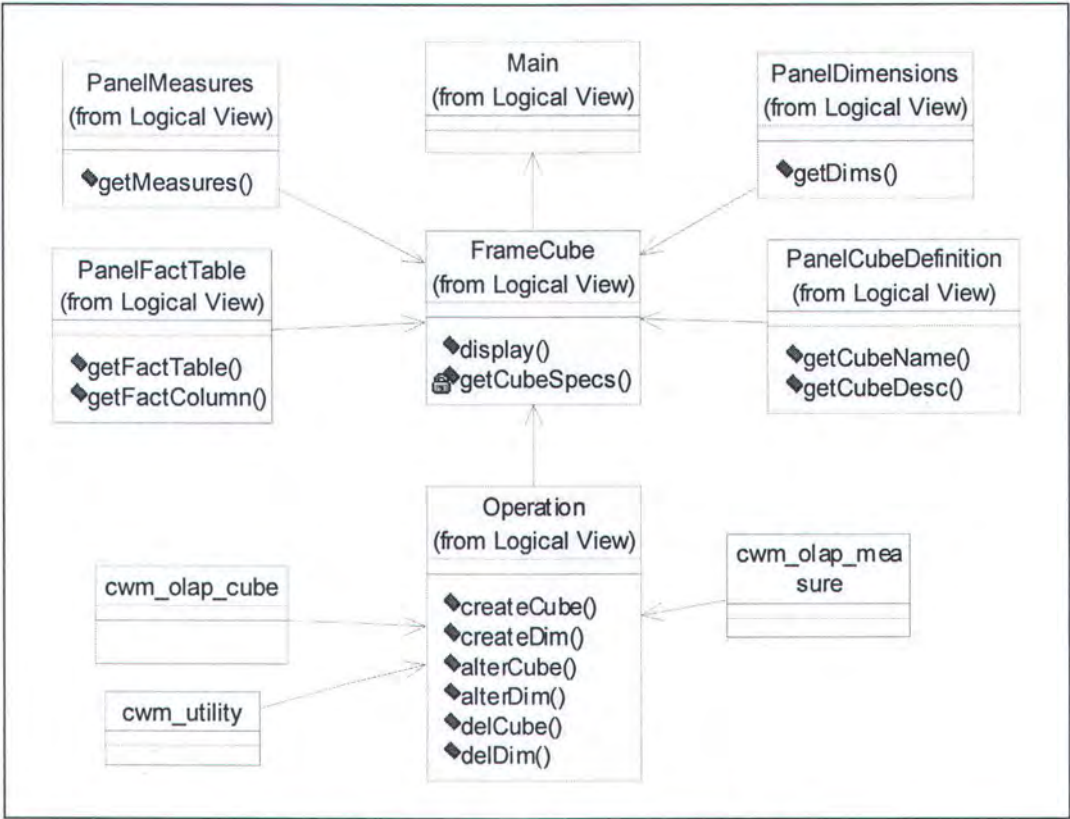
PanelDimTables digunakan untuk mendefinisikan tabel yang digunakan oleh dimensi. PanelDimLevels untuk mendefinisikan level-level yang menyusun dimensi. PanelDimHierarchy untuk menentukan *hierarchy* pada suatu dimensi. PanelDimJoin untuk membuat relasi pada dimensi jika yang dibuat adalah dimensi dengan tipe *snowflake*.

*Class* Operation digunakan untuk mengeksekusi perintah-perintah database sesuai dengan spesifikasi yang diinginkan user. Sedangkan *class-class* yang berawalan dengan *cwm\_* merupakan *package* pada Oracle OLAP yang dimanfaatkan pada operasi-operasi yang berhubungan dengan dimensi.

3.2.2 Obyek Pada Proses Cube

Operasi cube dilakukan pada *class* FrameCube. FrameCube menggabungkan *class* PanelCubeDefinition, PanelFactTable, PanelMeasures dan PanelDimensions. PanelCubeDefinition digunakan untuk mendapatkan informasi spesifikasi cube yang akan dibuat. PanelFactTable digunakan untuk mendapatkan data tabel fakta yang digunakan. PanelMeasures digunakan untuk pendefinisian *measure* pada cube. PanelDimensions digunakan untuk pendefinisian dimensi-dimensi yang terlibat pada cube.

*Class* Operation digunakan untuk mengeksekusi perintah-perintah database sesuai dengan keinginan user. Sedangkan *class-class* yang berawalan *cwm\_* merupakan *package* dari Oracle OLAP.

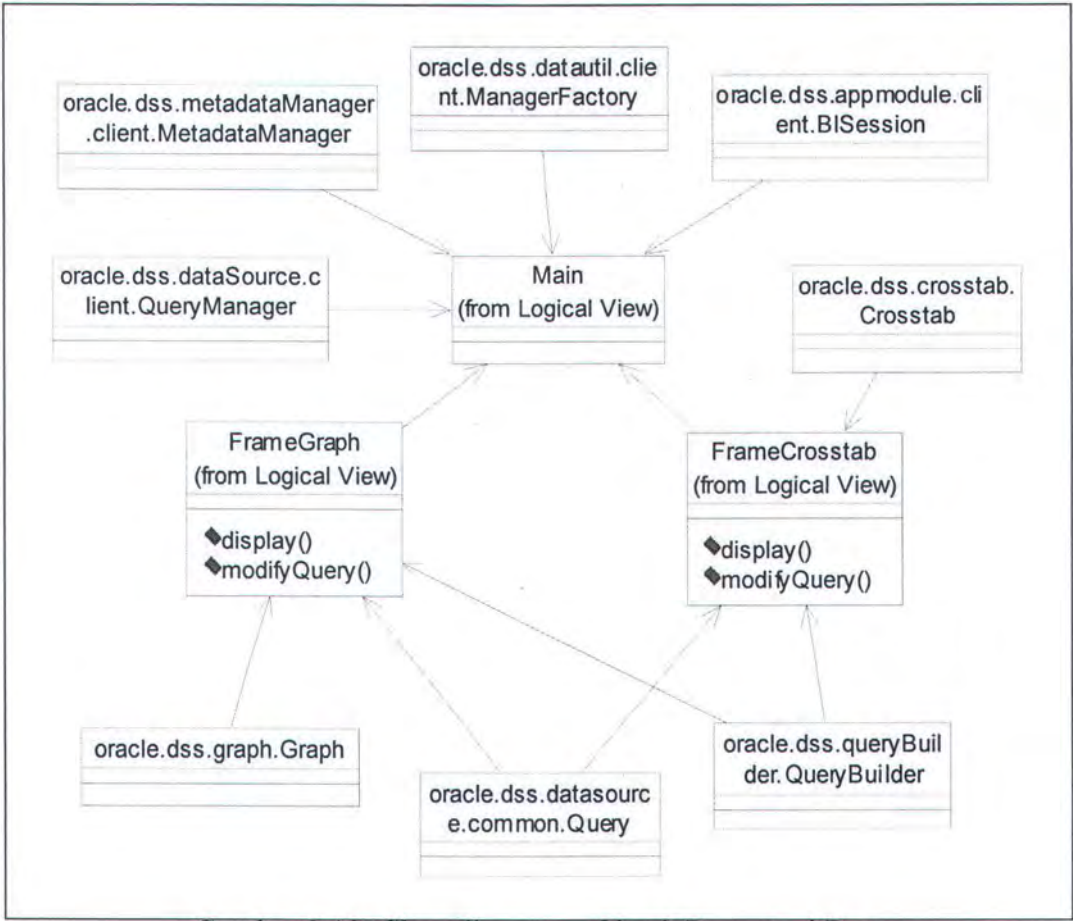


Gambar 3.18. Class Diagram – Obyek Cube



3.2.3 Obyek Pada Proses Presentasi Data

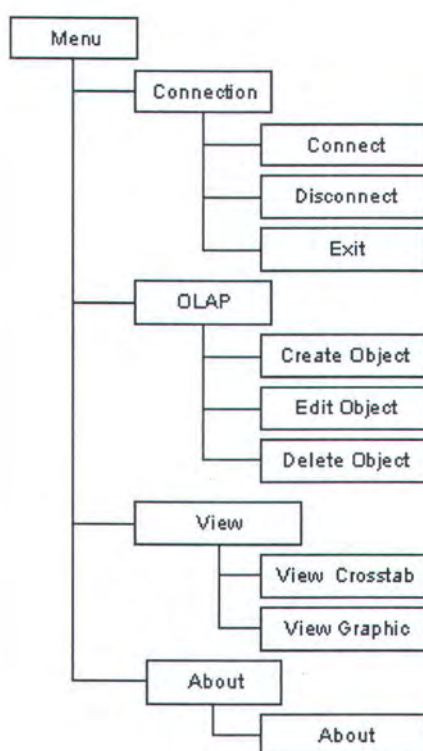
Ada 2 *class* JFrame yang digunakan yaitu FrameGraph dan FrameCrosstab untuk mempresentasikan data multidimensi dalam bentuk *crosstab* dan grafik. *Class-class* lain yang digunakan adalah MetadataManager, ManagerFactory dan BISession. Ketiga *class* ini digunakan untuk proses koneksi ke BI Beans. *Class* QueryManager dan Query digunakan untuk proses penarikan data sesuai dengan pilihan user. *Class* Crosstab dan Graph masing-masing digunakan untuk menampilkan data multidimensi dalam bentuk crosstab dan grafik. *Class* QueryBuilder digunakan untuk proses modifikasi *query* yang ditampilkan kepada user.



Gambar 3.19. Class diagram – Obyek Presentasi Data

### 3.3 Desain Antarmuka

Perangkat lunak ini menggunakan class JFrame pada Java untuk mengimplementasikan tampilan window. Didalamnya menggunakan konsep menu sebagai navigasi untuk membantu user dalam penggunaan perangkat lunak. Sebagai tambahan, juga disediakan *toolbar* yang menyediakan akses cepat untuk menjalankan operasi tertentu.



Gambar 3.20. Menu aplikasi

Menu-menu yang tersedia seperti terlihat pada gambar diatas, meliputi :

#### a. Menu Connection

Menu ini terdiri dari sub menu :

- Connect : digunakan untuk membuat koneksi database.

- Disconnect : digunakan apabila user menginginkan untuk memutuskan koneksi dengan database.
- Exit : digunakan untuk keluar dari perangkat lunak.

#### **b. Menu OLAP**

Menu ini terdiri dari sub menu :

- Create Object : sub menu ini digunakan apabila user ingin membuat metadata/obyek OLAP.
- Edit Object : sub menu ini digunakan apabila user ingin melakukan modifikasi metadata/obyek OLAP.
- Delete Object : sub menu ini digunakan apabila user ingin menghapus metadata/obyek OLAP.

#### **c. Menu View**

Menu ini terdiri dari sub menu :

- View Crosstab : sub menu ini digunakan apabila user ingin melihat data multidimensi dalam bentuk crosstab.
- View Graphic : sub menu ini digunakan apabila user ingin melihat data multidimensi dalam bentuk grafik.

#### **d. About**

Menu ini terdiri dari sub menu :

- About : digunakan untuk menampilkan dialog about yang berisi informasi singkat tentang perangkat lunak, versi dan pembuatnya.



Selain itu, penggunaan *context menu* juga diterapkan dalam perangkat lunak ini. Dengan menggunakan *context menu*, menu yang ditampilkan akan berubah sesuai dengan konteks obyek yang sedang dipilih user.

TO BELIEVE IS TO KNOW THAT WONDERFUL SURPRISES ARE  
JUST WAITING TO HAPPEN, AND ALL OUR HOPES AND  
DREAMS ARE WITHIN REACH.

## BAB IV

# IMPLEMENTASI PERANGKAT LUNAK

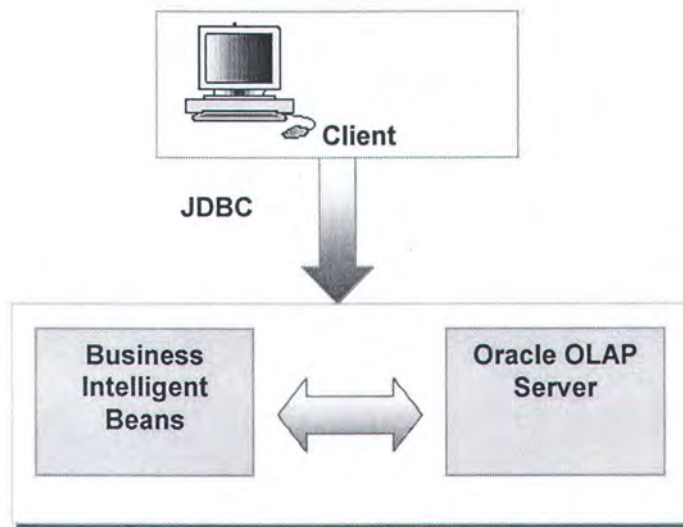
## BAB IV

### IMPLEMENTASI PERANGKAT LUNAK

Dalam bab ini akan dijelaskan mengenai proses implementasi pembuatan perangkat lunak. Pembuatan perangkat lunak ini menggunakan bahasa pemrograman Java dengan menggunakan Oracle JDeveloper sebagai lingkungan pembuatannya.

#### 4.1 Arsitektur Sistem

Perangkat lunak ini menggunakan model arsitektur *two-tier* seperti ditunjukkan pada gambar berikut :



Gambar 4.1. Arsitektur Sistem

Penggunaan model seperti ini mengakibatkan perangkat lunak yang berbasiskan Java berhubungan langsung dengan basisdata, terutama OLAP Server. Oleh karena itu perangkat lunak ini membutuhkan sebuah JDBC driver



yang bisa berkomunikasi dengan *Database Management System* (DBMS) yang diakses.

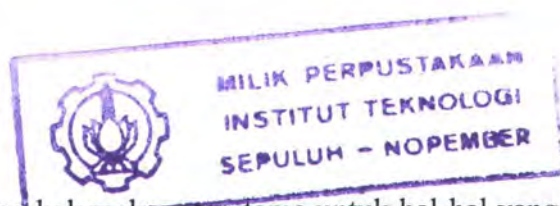
JDBC (Java Database Connectivity) didefinisikan sebagai sebuah antarmuka untuk berelasi dengan sumber data. JDBC memiliki kemampuan mengeksekusi perintah-perintah SQL untuk mengakses dan mengelola basisdata relasional. JDBC didesain berorientasi obyek, *Application Programming Interface* (API) untuk mengakses basisdata yang berbasiskan Java, dan ditujukan sebagai standar oleh pengembang aplikasi Java dan perusahaan pembuat basisdata. Pemakaian Java dan JDBC ini mengakibatkan pengembang aplikasi tidak perlu membuat aplikasi yang berbeda untuk sistem operasi yang berbeda.

Untuk terhubung dengan server Oracle, perangkat lunak ini menggunakan Oracle Thin JDBC. Driver ini merupakan driver yang menggunakan soket Java untuk melakukan koneksi secara langsung ke Oracle. Karena driver ini secara keseluruhan dibuat menggunakan Java, maka driver ini bersifat *platform-independent*.

#### 4.2 Tahap Persiapan

Oracle9i release 2 memiliki beberapa kelemahan, terutama untuk hal-hal yang berkaitan dengan data warehouse dan OLAP [DAM-02]. Untuk meminimalisasi beberapa kelemahan ini, maka harus diinstal patch 9.2.0.2.1 yang dirilis Oracle (lampiran).

Setelah proses instalasi patch selesai, ada beberapa tugas pasca instalasi yang harus dilakukan. Patch tidak akan bekerja dengan baik, jika operasi-operasi berikut ini tidak dilakukan.



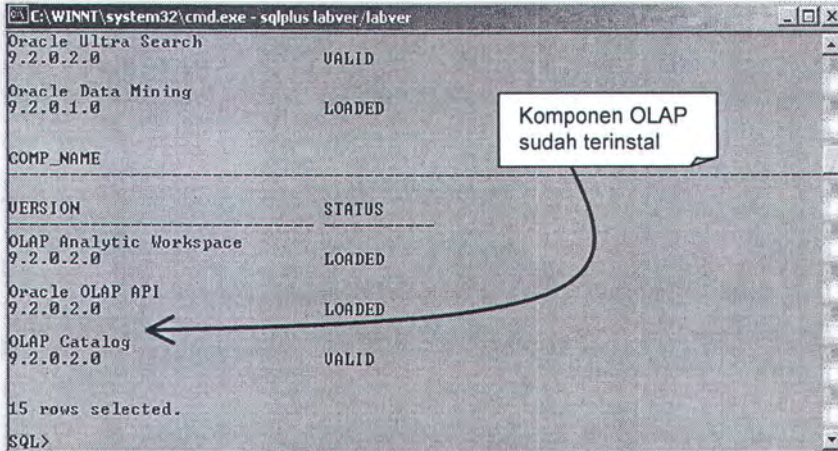
```

SQL> shutdown immediate
SQL> startup migrate
SQL> spool patch.log
SQL> @/$Oracle_Home/rdbms/admin/catpatch.sql
SQL> spool off
SQL> shutdown
SQL> startup
SQL> @/$Oracle_Home/rdbms/admin/utlrp.sql

```

Untuk mengecek apakah patch sudah terinstall dengan benar, dapat digunakan perintah SQL berikut. Dengan mengeksekusi perintah tersebut, hasil yang diperoleh seperti pada gambar.

```
SQL> select comp_name, version, status from dba_registry
```



COMP_NAME	VERSION	STATUS
Oracle Ultra Search	9.2.0.2.0	VALID
Oracle Data Mining	9.2.0.1.0	LOADED
OLAP Analytic Workspace	9.2.0.2.0	LOADED
Oracle OLAP API	9.2.0.2.0	LOADED
OLAP Catalog	9.2.0.2.0	VALID

15 rows selected.

SQL>

Gambar 4.2. Mengecek kevalidan komponen OLAP.

Selanjutnya, perlu juga diinstal Oracle BI Beans pada direktori ORACLE\_HOME (lampiran). Hal ini perlu dilakukan agar komponen-komponen BI Beans dapat dimanfaatkan oleh server Oracle.

## 4.3 Implementasi Proses

### 4.3.1 Pembuatan Koneksi Oracle

Perangkat lunak ini mempunyai dua fungsi koneksi yaitu, koneksi ke Oracle OLAP dan koneksi ke Oracle BI Beans. Masing-masing koneksi dijelaskan sebagai berikut.

#### 4.3.1.a Koneksi Oracle OLAP

Koneksi ini digunakan ketika user pertama kali *log-in* ke perangkat lunak. Semua operasi OLAP akan menggunakan tipe koneksi ini. Berikut ini adalah *procedure* untuk koneksi ke Oracle OLAP.

```
protected void connect(String username, String
password, String sid){
    String url = "jdbc:oracle:thin:@localhost:1521:" +
    sid;

    try {
        DriverManager.registerDriver(new OracleDriver());
        conn = (OracleConnection)
        DriverManager.getConnection(url, username, password);

        } catch(Exception e){
            System.out.println("Error encountered : " +
            e.getMessage());
        }
    }
```

#### 4.3.1.b Koneksi Oracle BI Beans

Koneksi ini akan digunakan ketika user melakukan operasi *view* data multidimensi. Berikut ini adalah *procedure* untuk koneksi ke Oracle BI Beans.



```

protected void bi_connect(String user,String pass){
    session = new BISession(configFile);
    session.setBIUser(new BIUser(user));

    try {
        ManagerFactory factory =
session.getManagerFactory();

        metaManager =
(MetadataManager) factory.lookupManager(
ManagerFactory.METADATA_MANAGER, null, true);
        qryManager = (QueryManager) factory.lookupManager(
                                ManagerFactory.QUERY_MANAGER,
null, true);
        persManager =
(PersistenceManager) factory.lookupManager(
ManagerFactory.PERSISTENCE_MANAGER, null, true);

    }
    catch (BISessionException bse) {
        ErrorBox eb = new ErrorBox(bse.getMessage());
    }
    catch (MetadataManagerException mme) {
        ErrorBox eb = new ErrorBox(mme.getMessage());
    }
    catch (Exception ex) {
        ErrorBox eb = new ErrorBox(ex.getMessage());
    }
}

```

### 4.3.2 CWM Package Pada Oracle

Common Warehouse Metadata (CWM) merupakan *repository* standar yang digunakan oleh Oracle Data Warehouse. CWM *repository* skema adalah produk yang berdiri sendiri yang dapat dimanfaatkan secara bersama-sama oleh produk lain [KEL-02]. CWM berisi *package* yang didalamnya berisi fungsi-fungsi untuk mengolah metadata OLAP. Daftar *package* tersebut dapat dilihat pada tabel dibawah ini.

Table 4.1. Daftar Package Oracle OLAP

Nama Package	Keterangan
CWM_OLAP_CUBE	Package berisi fungsi-fungsi yang berkenaan dengan penanganan cube.
CWM_OLAP_MEASURE	Package berisi fungsi-fungsi penanganan masure
CWM_OLAP_DIMENSION	Package berfungsi untuk mengolah data dimensi
CWM_OLAP_HIERARCHY	Package berfungsi untuk mengolah hierarchy pada dimension
CWM_OLAP_LEVEL	Package yang berkenaan dengan penanganan level.
CWM_OLAP_DIM_ATTRIBUTUTE	Package yang berhubungan dengan pengolahan atribut dimension
CWM_OLAP_LEVEL_ATTRIBUTE	Package yang bertugas mengolah data atribut level.
CWM_CLASSIFY	Package yang berfungsi untuk mengolah data pada katalog

*Package* CWM ini akan digunakan pada class *Operation* untuk proses pengolahan metadata OLAP. Tiap *package* memiliki fungsi dan kegunaan masing-masing. Sebagai contoh, untuk membuat metadata OLAP berupa dimensi maka *package* akan digunakan adalah *cwm\_olap\_dimension*, *cwm\_olap\_level*, *cwm\_olap\_hierarchy*, *cwm\_olap\_dim\_attribute*, dan *cwm\_olap\_level\_attribute*.

4.3.3 Class Operation

Class yang digunakan untuk mengeksekusi operasi-operasi OLAP adalah *Operation*. Berikut ini adalah *method summary* dari class *Operation*.

Tabel 4.2. Metode pada class Operation

Tipe Metode	Nama Metode dan Kegunaan
void	<b>createCube</b> (String skema, String cubeName, String factTable, Vector dims, Vector measure) Procedure untuk membuat cube.



void	<b><u>createDimension</u></b> (String schema, String dimName, Vector tables, Vector levels, Vector hiers, Vector structure) Procedure untuk membuat dimensi.
java.util.Vector	<b><u>cubeDetails</u></b> (String schema,String cubename) Procedure ini digunakan untk mendapatkan cube properties
Void	<b><u>deleteCube</u></b> (String cubeName) Untuk Menghapus cube dari database.
Void	<b><u>deleteDimension</u></b> (String dimname) Untuk menghapus dimensi.
java.lang.String	<b><u>determineLowestLevel</u></b> (Vector struktur) Fungsi ini digunakan untuk menentukan level paling rendah dari sejumlah struktur hierarki.
java.util.Vector	<b><u>dimDetails</u></b> (String schema, String dimname) Fungsi ini digunakan untuk mendapatkan dimension properties
java.lang.String	<b><u>getDefaultHierarchy</u></b> (String owner,String dimension) Fungsi ini digunakan untuk mendapatkan default hirarki dari suatu dimensi
java.lang.String	<b><u>getDimensionLowestLevel</u></b> (String skema, String dimension_name) Fungsi ini digunakan untuk mendapatkan level paling bawah dari sebuah dimensi
java.lang.String	<b><u>getFactTableForeignKey</u></b> (String skema, String dimension_name) Fungsi untuk mendapatkan foreign key dari fact table yang mengacu pada dimensi yang disebutkan pada parameter
java.lang.String	<b><u>getPrimaryKeyColumn</u></b> (String schema, String table_name) Fungsi untuk mendapatkan primary key dari suatu table.
java.lang.String	<b><u>getTablePrimaryKey</u></b> (String schema, String table_name) Procedure untuk mendapatkan primary key dari suatu table.
java.util.Vector	<b><u>levelDetails</u></b> (String schema, String dimname) Procedure untuk menarik informasi level properties
void	<b><u>olapMetadataRefresh</u></b> () Procedure ini untuk me-refresh metadata olap, digunakan setiap selesai membuat satu obyek olap.
java.util.Vector	<b><u>selectAttributes</u></b> (String schema, String dim_name, String level name)



	Untuk memilih attribute dari suatu dimensi.
java.util.Vector	<b><u>selectDimensions</u></b> (String schema)
java.util.Vector	<b><u>selectLevels</u></b> (String schema, String dim_name) Procedure untuk memilih level dari suatu dimensi.

#### 4.3.4 Class HierarchyDim

HierarchyDim adalah *class* yang digunakan untuk memodelkan metadata OLAP berupa *hierarchy*. Berikut ini adalah metode-metode yang ada pada *class* HierarchyDim.

Tabel 4.3. Metode pada class HierarchyDim

Tipe Metode	Nama Metode dan Kegunaan
void	<b><u>addRelation</u></b> (Relation rel) Procedure untuk menambah relasi
java.lang.String	<b><u>getHierName</u></b> () Procedure untuk mendapatkan nama hierarki
java.lang.String	<b><u>getLevelTopDown</u></b> (int idx) Procedure untuk mendapatkan level pada struktur
int	<b><u>getStructureSize</u></b> () Procedure untuk mendapatkan ukuran struktur
java.util.Vector	<b><u>getVectorRelation</u></b> () Procedure untuk mendapatkan info relasi
java.util.Vector	<b><u>getVectorStructure</u></b> () Procedure untuk mendapatkan struktur hierarchy dalam bentuk vector
boolean	<b><u>haveRelation</u></b> () Procedure untuk mengetahui apakah hierarchy mempunyai relasi
void	<b><u>populateStructure</u></b> (java.util.Vector levels) Procedure untuk mengisi struktur hierarki
void	<b><u>setHierName</u></b> (java.lang.String hier_name) Procedure untuk menset nama hierarchy
void	<b><u>setRelation</u></b> (java.lang.String level_parent, java.lang.String level_child)

	Procedure untuk menset relasi pada hierarchy
--	--

#### 4.3.5 Class LevelDim

*Class* LevelDim merupakan representasi metadata OLAP berupa *level*. Berikut ini metode-metode yang terdapat pada *class* LevelDim.

Tabel 4.4. Metode pada class LevelDim

Tipe Metode	Nama Metode dan Kegunaan
java.util.Vector	<b><u>getAttributes()</u></b> Fungsi untuk mengambil informasi attribute dalam bentuk Vector
java.lang.String	<b><u>getLevelColumn()</u></b> Procedure untuk mendapatkan kolom level
java.lang.String	<b><u>getLevelName()</u></b> Procedure untuk mendapatkan nama level
java.lang.String	<b><u>getLevelTable()</u></b> Procedure untuk mendapatkan table dari level
java.lang.String	<b><u>getTableAndColumn()</u></b> Fungsi untuk mendapatkan table dan kolom level dalam format bersambung
boolean	<b><u>isLowestLevel()</u></b> Procedure untuk mendapatkan informasi apakah level ini adalah level terbawah
void	<b><u>setLevelColumn()</u></b> (java.lang.String col) Procedure untuk mengeset kolom level
void	<b><u>setLevelName()</u></b> (java.lang.String name) Procedure untuk mengeset nama level
void	<b><u>setLevelTable()</u></b> (java.lang.String tab) Procedure untuk mengeset tabel level
void	<b><u>setLowestLevel()</u></b> (boolean boo) Procedure untuk mengeset level terbawah
void	<b><u>setTableAndColumn()</u></b> (java.lang.String tab_col) Procedure untuk mengeset tabel dan kolom level sekaligus



4.3.6 Class Relation

Class Relation digunakan untuk merepresentasikan relasi antara dua level pada dua tabel yang membentuknya. Relasi terbentuk jika operasi yang dilakukan adalah pembuatan metadata OLAP dimensi bertipe *snowflake*. Berikut ini adalah metode-metode pada *class* Relation.

Tabel 4.5. Metode pada class Relation

Tipe Metode	Nama Metode dan Kegunaan
java.lang.String	<u><b>getColumnChild()</b></u> Mendapatkan kolom child
java.lang.String	<u><b>getColumnParent()</b></u> Mendapatkan kolom parent
java.lang.String	<u><b>getLevelChild()</b></u> Mendapatkan level child
java.lang.String	<u><b>getLevelParent()</b></u> Mendapatkan level parent
java.lang.String	<u><b>getTableChild()</b></u> Mendapatkan tabel child
java.lang.String	<u><b>getTableParent()</b></u> Mendapatkan tabel parent
void	<u><b>setColumn</b></u> (String column_parent, String column_child) Mengeset kolom parent dan child.
void	<u><b>setColumnChild</b></u> (String col) Mengeset kolom child
void	<u><b>setColumnParent</b></u> (String col) Mengeset kolom parent
void	<u><b>setTable</b></u> (String table_parent, String table_child) Mengeset tabel parent dan child

4.3.7 Attribute Long\_Description Dan Short\_Description

Pada setiap pembuatan metadata *dimension*, OLAP Catalog memerlukan suatu informasi khusus dalam bentuk *attribute*. *Attribute* ini dinamakan



Long\_Description dan Short\_Description. *Attribute* ini harus didefinisikan untuk setiap level yang dibuat pada suatu metadata *dimension*. Selanjutnya, attribute ini akan digunakan pada proses *view* metadata, baik dalam bentuk *crosstab* maupun *grafik*.

Untuk memudahkan user dalam pembuatan attribute ini, maka aplikasi OLAP Builder akan melakukan pembuatan attribute secara otomatis, sehingga user tidak perlu melakukan pendefinisian secara manual. Semua proses ini dilakukan tanpa sepengetahuan user. Berikut ini adalah potongan program untuk mendefinisikan attribute Long\_Description dan Short\_Description.

```
public void createDimension(String schema,String dimName,
Vector tables,Vector levels, Vector hiers, Vector
structure){

...

//deklarasi attribute Long_Description dan
//Short_Description
newDimName = dimName.toUpperCase();
cwm_define_short_desc1 =
"cwm_olap_dim_attribute.create_dimension_attribute(\"" +
schema + "\",\"" + newDimName +
"\",\"Long_Description\", \"Long_Description\", \"\"); ";

cwm_define_short_desc1 =
"cwm_olap_dim_attribute.create_dimension_attribute(\"" +
schema + "\",\"" + newDimName +
"\",\"Short_Description\", \"Short_Description\", \"\"); ";

cwm_define_short_desc2 =
"cwm_classify.add_entity_descriptor_use(41, " +
"cwm_utility.DIMENSION_ATTRIBUTE_TYPE,\"" + schema +
"\",\"" + newDimName +
"\",\"Long_Description\"); ";

cwm_define_short_desc2 =
"cwm_classify.add_entity_descriptor_use(42, " +
"cwm_utility.DIMENSION_ATTRIBUTE_TYPE,\"" + schema +
"\",\"" + newDimName +
"\",\"Short_Description\"); ";
```

```

//definisi attribute untuk tiap level
LevelDim temp = (LevelDim)levels.elementAt(i);
for(int i=0; i<levels.size(); i++){
    attr_short_desc1 = attr_short_desc1 +
    "cwm_olap_dim_attribute.add_level_attribute(\"'\" + schema
    + "\",\"" + newDimName + "\",\"" + "Long_Description"
    + "\",\"" + temp.getLevelName() + "\",\"" +
    attrCol + "\"); ";

    attr_short_desc1 = attr_short_desc1 +
    "cwm_olap_dim_attribute.add_level_attribute(\"'\" + schema
    + "\",\"" + newDimName + "\",\"" + "Short_Description"
    + "\",\"" + temp.getLevelName() + "\",\"" +
    attrCol + "\"); ";

    attr_short_desc2 = attr_short_desc2 +
    "cwm_classify.add_entity_descriptor_use(41,
    cwm_utility.LEVEL_ATTRIBUTE_TYPE,\"'\" + schema + "\",\"" +
    newDimName + "\",\"" + temp.getLevelName() + "\",\"" +
    attrCol + "\"); ";

    attr_short_desc2 = attr_short_desc2 +
    "cwm_classify.add_entity_descriptor_use(42,
    cwm_utility.LEVEL_ATTRIBUTE_TYPE,\"'\" + schema + "\",\"" +
    newDimName + "\",\"" + temp.getLevelName() + "\",\"" +
    attrCol + "\"); ";

} //end for

...

//eksekusi attribute
try{
    cs = conn.prepareStatement("begin " +
    cwm_define_short_desc1 + " " +
    cwm_define_short_desc2 + " " +
    "commit; end;");
    cs.execute();
    status = STATUS_SUCCESS;
} catch (SQLException e) {
    status = STATUS_FAILED;
    errormsg = e.getMessage();
}

```

### 4.3.8 Metadata Refresh

Pada setiap operasi OLAP yang dilakukan, database harus di *refresh* agar informasi mengenai perubahan metadata dapat segera diupdate. Proses ini harus

dilakukan agar metadata tetap valid dan tidak mengganggu operasi OLAP selanjutnya. Untuk *me-refresh* metadata seorang user harus terkoneksi pada database menggunakan skema OLAPSYS kemudian mengeksekusi perintah berikut :

```
SQL> EXEC CWM2_OLAP_METADATA_REFRESH.MR_REFRESH
```

Agar tidak menyulitkan user melakukan operasi diatas, maka perangkat lunak ini menyediakan procedure metadata refresh yang akan dipanggil secara otomatis begitu perubahan metadata terjadi. Berikut ini adalah procedure metadata refresh.

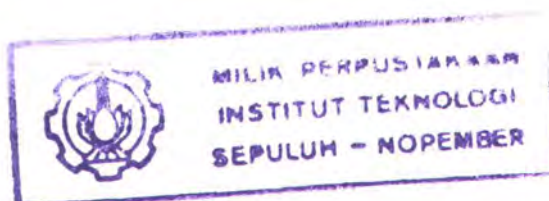
```
public void olapMetadataRefresh(){
    CallableStatement cs;
    try{
        cs = conn.prepareCall("begin " +
            "cwm2_olap_metadata_refresh.mr_refresh; " +
            "end;");
        cs.execute();

    } catch (SQLException e){
        System.out.println("error : " +
            e.getMessage());
    }
}
```

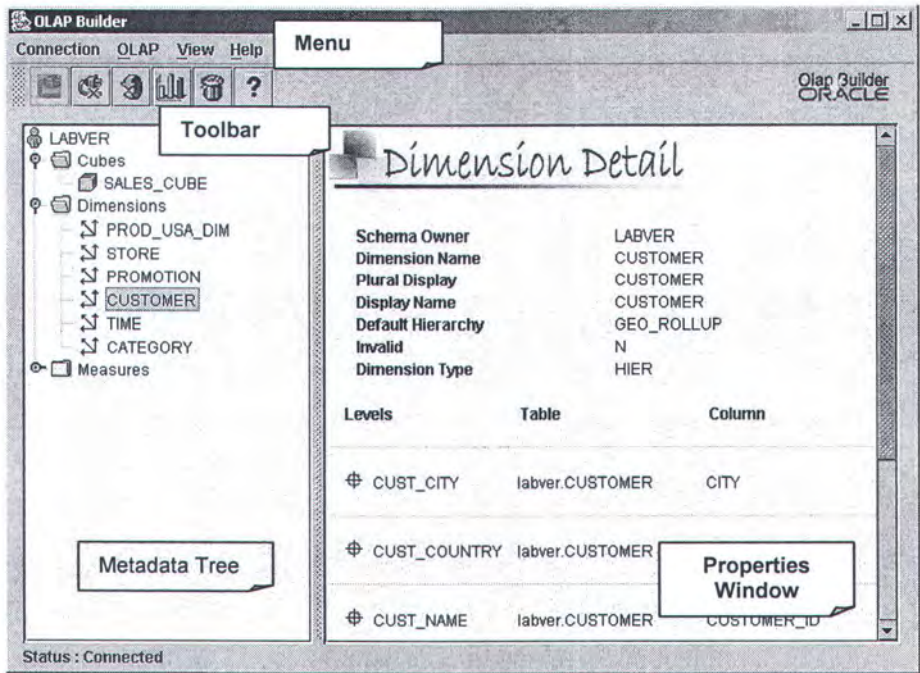
#### 4.4 Implementasi Antarmuka

Sesuai penjelasan pada bab 3.3, perangkat lunak ini akan mengimplementasikan antarmuka dengan sistem *window*. Didalamnya terdapat menu, *context menu* dan *toolbar* untuk mempermudah user berinteraksi dengan perangkat lunak.

Perangkat lunak ini juga memanfaatkan sistem *dialog* untuk berinteraksi langsung dengan user. Berikut ini adalah gambar tampilan utama perangkat lunak.





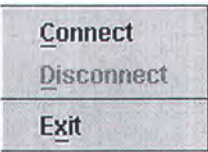


Gambar 4.3. Tampilan utama perangkat lunak

Sebelum dilakukan koneksi, seluruh menu dan *toolbar* pada tampilan utama (*Main*) dalam kondisi *disable*. Status koneksi dapat diketahui dari *label* pada bagian bawah window yang menyatakan bahwa koneksi belum tersambung.

4.4.1 Menu Connection

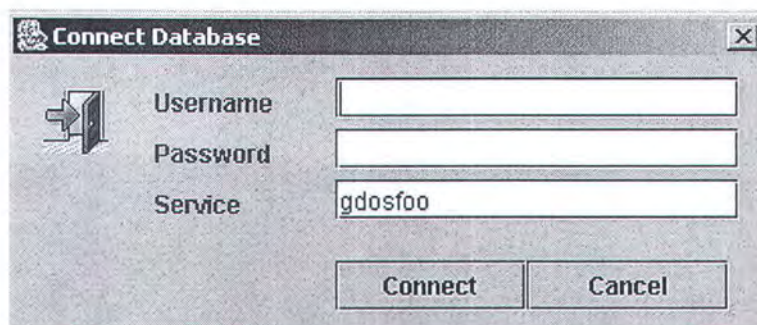
Menu *connection* terdiri dari 3 sub menu, yaitu Connect, Disconnect dan Exit.



Gambar 4.4. Sub Menu Connection

#### 4.4.1.a Sub Menu Connect

Sub menu connect digunakan untuk melakukan koneksi ke database. Jika menu ini diklik maka akan memunculkan kotak *dialog* koneksi seperti ditunjukkan gambar dibawah ini.



Gambar 4.5. Dialog untuk melakukan koneksi

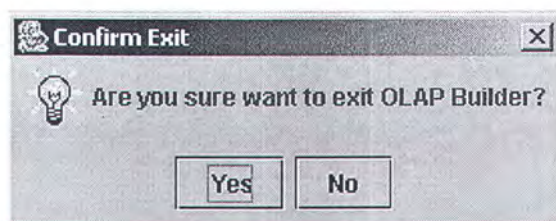
Untuk melakukan koneksi, user harus mengisi informasi berupa username, password dan service ID dari database. Tombol Connect untuk melanjutkan proses koneksi, sedangkan tombol Cancel untuk membatalkan proses.

#### 4.4.1.b Disconnect

Sub menu Disconnect digunakan untuk memutus koneksi dari database. Jika menu ini diklik, maka seluruh tampilan window utama akan dihapus dan akan tampak seperti tampilan sebelum koneksi.

#### 4.4.1.c Exit

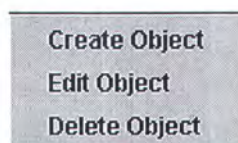
Sub menu Exit digunakan untuk keluar dari aplikasi. Jika sub menu ini diklik, akan muncul kotak *dialog* konfirmasi kepada user untuk menanyakan akan benar-benar akan menutup aplikasi. Gambar berikut menunjukkan efek tersebut.



Gambar 4.6. Dialog konfirmasi keluar aplikasi

#### 4.4.2 Menu OLAP

Menu OLAP terdiri dari 3 sub menu, yaitu Create Object, Edit Object dan Delete Object.



Gambar 4.7. Sub Menu OLAP

##### 4.4.2.a Sub Menu Create Object

Jika sub menu ini diklik, maka efek yang akan dihasilkan adalah muncul form pembuatan obyek OLAP. Form yang akan muncul tergantung dari konteks obyek yang sedang aktif, apakah form untuk membuat dimensi (*FrameDimension*) atau form untuk membuat cube (*FrameCube*). Informasi detail mengenai *FrameDimension* dan *FrameCube* akan dijelaskan kemudian.

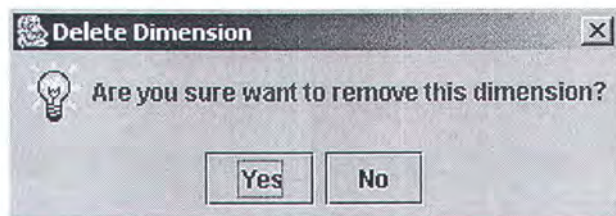
##### 4.4.2.b Sub Menu Edit Object

Efek yang akan terjadi jika sub menu ini diklik adalah, munculnya form modify metadata dimension.



#### 4.4.2.c Sub Menu Delete Object

Jika sub menu ini diklik, maka akan muncul *dialog* konfirmasi apakah obyek benar-benar akan dihapus. Hal ini ditunjukkan pada gambar dibawah.



Gambar 4.8. Dialog konfirmasi menghapus obyek

Jika user mengklik tombol Yes, maka proses penghapusan akan diteruskan. Namun jika tombol No yang diklik, proses penghapusan akan dibatalkan.

#### 4.4.3 Menu View

Menu View terdiri dari 2 sub menu, yaitu View Crosstab dan View Graphic. Hal ini ditunjukkan pada gambar berikut.



Gambar 4.9. Sub Menu View

##### 4.4.3.a Sub Menu View Crosstab

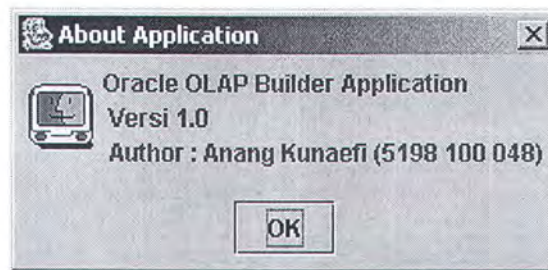
Sub menu View Crosstab digunakan untuk melihat data multidimensi dalam bentuk Crosstab. Jika sub menu ini diklik maka akan muncul *FrameCrosstab* yang akan menampilkan data multidimensi sesuai dengan *measure* atau *cube* yang dipilih.

##### 4.4.3.b Sub Menu View Graphic

Jika sub menu View Graphic diklik, maka akan muncul *FrameGraphic* yang akan menampilkan data multidimensi dalam bentuk Grafik.

#### 4.4.4 Menu About

Menu About hanya memiliki satu sub menu yaitu, about. Jika sub menu about diklik, maka akan muncul *dialog* seperti gambar berikut. *Dialog* ini berisi informasi mengenai perangkat lunak dan pembuatnya.



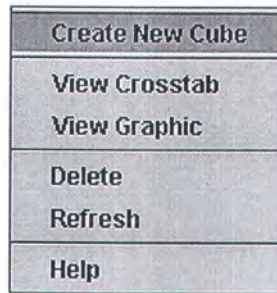
Gambar 4.10. Dialog about

#### 4.4.5 Context Menu

*Context* menu adalah menu yang dapat berubah-ubah isinya sesuai dengan obyek yang sedang dipilih oleh user. Untuk mengaktifkan *context* menu, user mengklik kanan tombol pada mouse. Pada perangkat lunak ini, *context* menu digunakan selain untuk mempermudah user, juga untuk menginformasikan kepada user aktivitas/proses apa saja yang bisa dilakukan berkaitan dengan obyek tertentu.

##### 4.4.5.a Context Menu Pada Obyek Cube

Jika obyek *cube* diklik kanan, maka akan muncul *context menu* seperti gambar berikut.

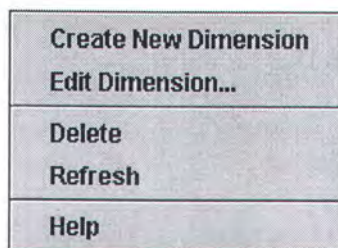


Gambar 4.11. Context menu untuk obyek cube.

Sesuai dengangambar, menu ini memiliki 5 buah item menu yaitu, Create New Cube, View Crosstab, View Graphic, Delete, Refresh dan Help. Jika menu item ini diklik maka akan memberikan efek yang sama dengan ketika user memilih lewat menu. Sebagai contoh, memilih Create New Cube pada *context menu* sama saja dengan memilih menu OLAP > Create Object.

#### 4.4.5.b Context Menu Pada Obyek Dimension

Jika obyek *dimension* diklik kanan, maka akan memunculkan *context* menu seperti gambar berikut.



Gambar 4.12. Context menu untuk obyek dimensi



#### 4.4.6 Form FrameDimension

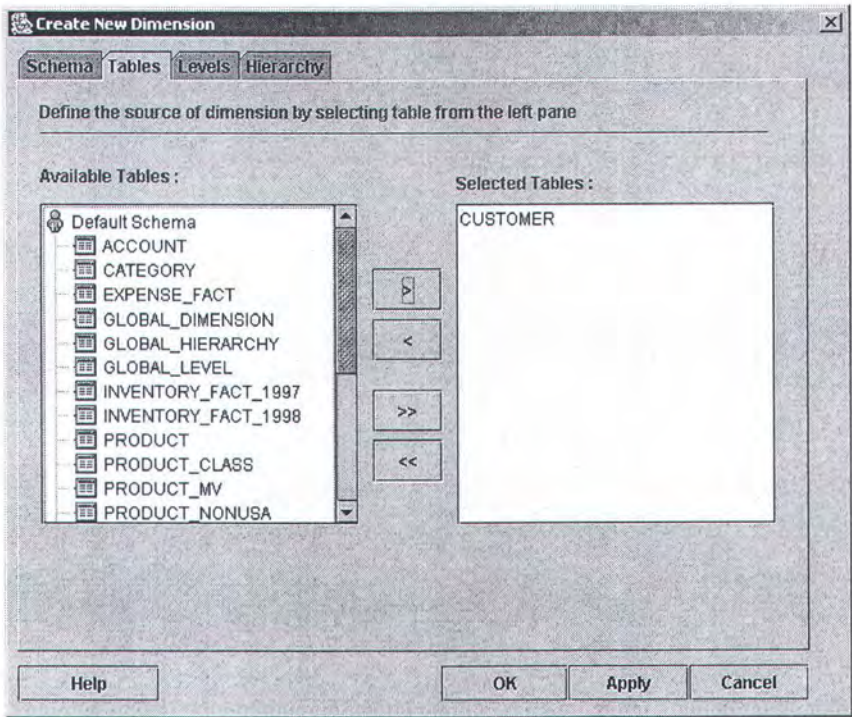
Form *frmDimension* digunakan untuk membuat dan mengedit obyek dimensi. Form ini mengimplementasikan class *JTabbedPane* pada Java untuk membuat tab. Form ini memiliki 4 buah tab yaitu, Schema, Tables, Levels dan Hierarchies. Masing-masing tab akan dijelaskan sebagai berikut.

##### 4.4.6.a FrameDimension – Tab Schema

Gambar 4.13. Tab Schema pada FrameDimension

Pada tab ini terdapat *text field* untuk mendefinisikan nama dimensi yang akan dibuat. Dua buah grup *radio button* untuk menentukan skema dimensi dan tipe dimensi. Pemilihan salah satu *radio button* pada tab ini akan mempengaruhi tampilan pada tab selanjutnya.

4.4.6.b FrameDimension – Tab Tables



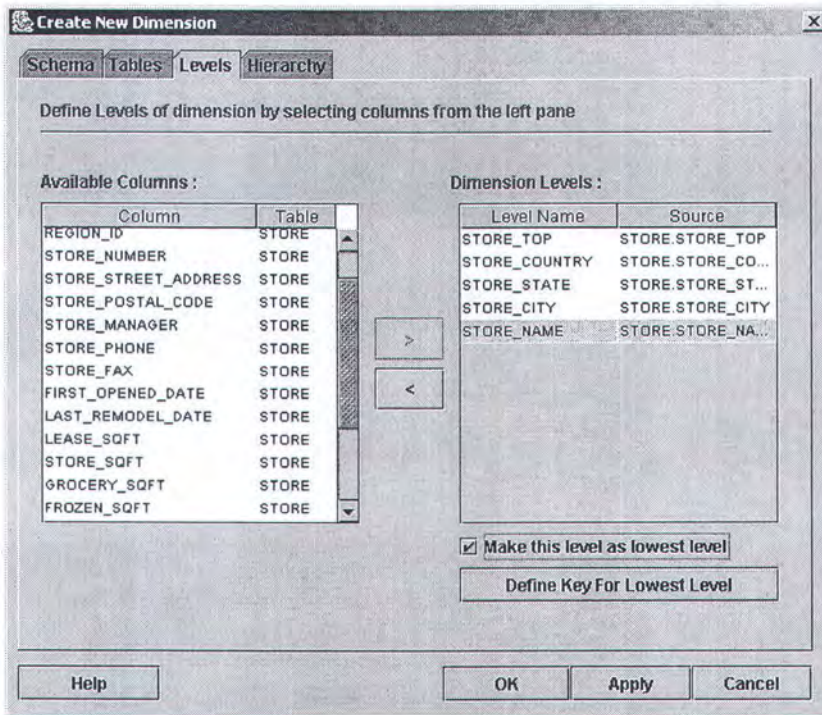
Gambar 4.14. Tab Tables pada FrameDimension

Pada tab ini terdapat komponen *JTree* untuk menampilkan daftar tabel yang terdapat pada suatu skema. Komponen *JList* pada bagian kanan digunakan untuk menampilkan tabel-tabel yang dipilih oleh user. Empat buah komponen *JButton* digunakan untuk menambah dan menghapus tabel dari daftar.

Aktivitas yang dilakukan user pada tab ini adalah memilih tabel yang akan menjadi sumber data dari dimensi yang akan dibuat. Jika pada tab *Schema* user memilih *star schema*, maka pada tab ini user hanya boleh memilih satu tabel. Namun, jika yang dipilih adalah *snowflake*, maka user boleh memilih lebih dari satu tabel.



#### 4.4.6.c FrameDimension – Tab Levels



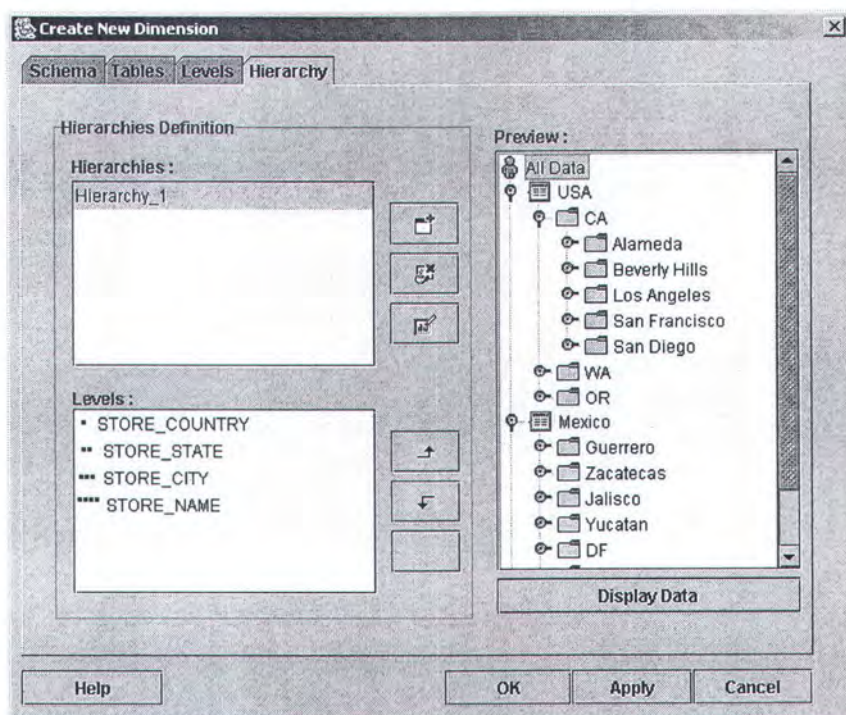
Gambar 4.15. Tab Levels pada FrameDimension

Pada tab ini user menentukan level dimensi berdasarkan kolom-kolom yang ada pada tabel yang telah dipilih pada tab sebelumnya. Level merupakan salah satu obyek yang membentuk dimensi. Obyek level ini akan digunakan untuk membentuk struktur hirarki pada dimensi.

Tab ini mengimplementasikan dua komponen *JTable*. Masing-masing tabel terdiri dari dua kolom. Tabel kiri menampilkan daftar kolom pada tabel yang bisa dipilih. Tabel kanan menunjukkan level yang telah dipilih user. Juga terdapat 2 komponen *JButton* untuk memilih dan menghapus level. Komponen *JCheckBox* digunakan untuk menandai apakah suatu level adalah *lowest level*. Serta tambahan 1 *button* untuk menentukan kolom *key* pada *lowest level*.



#### 4.4.6.d FrameDimension – Tab Hierarchies

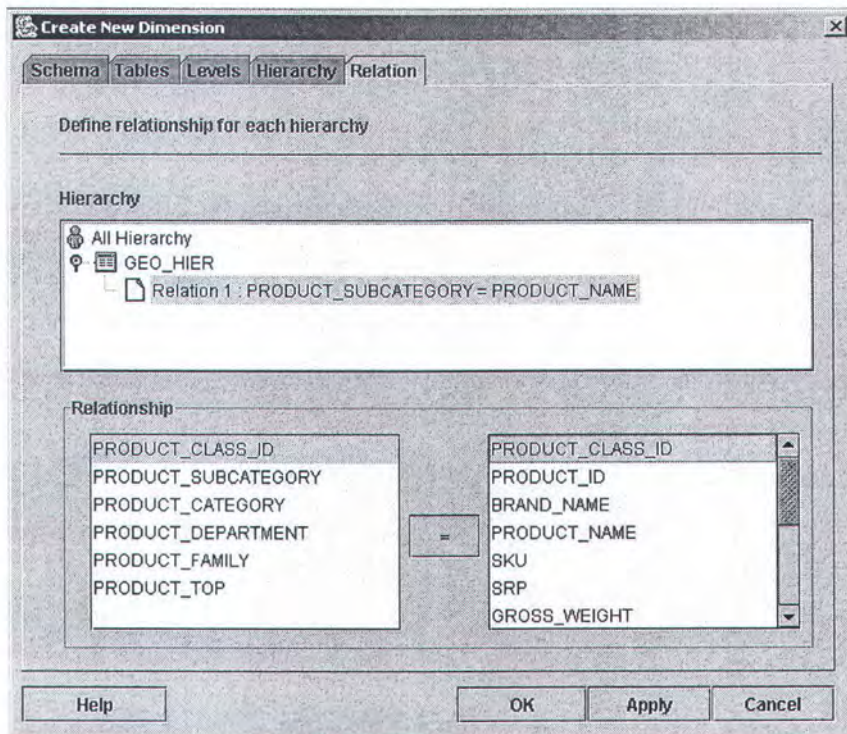


Gambar 4.16. Tab Hierarchies pada FrameDimension

Pada tab *hierarchies* user menentukan struktur hirarki yang akan digunakan pada dimensi. Struktur hirarki terdiri dari level-level yang ditentukan user pada tab sebelumnya.

Pada tab ini terdapat dua komponen *JList* yang masing-masing menampilkan nama hirarki dan struktur level yang dimilikinya. Pada bagian kanan terdapat komponen *JTree* yang akan menampilkan *preview data* dari dimensi yang akan dibuat berdasarkan struktur hirarki yang ada.

#### 4.4.6.c FrameDimension – Tab Relation



Gambar 4.17. Tab Relation pada FrameDimension

Pada tab *Relation* user menentukan relasi antara level-level yang berhubungan, jika dimensi yang dibuat adalah *snowflake*.

Pada tab ini terdapat dua komponen *JList* untuk menampung dua tabel yang saling berelasi, serta komponen *JTree* untuk menampilkan daftar hierarchy dan level yang saling berelasi. Komponen *JButton* tidak mempunyai fungsi selain untuk menampilkan tanda sama dengan.

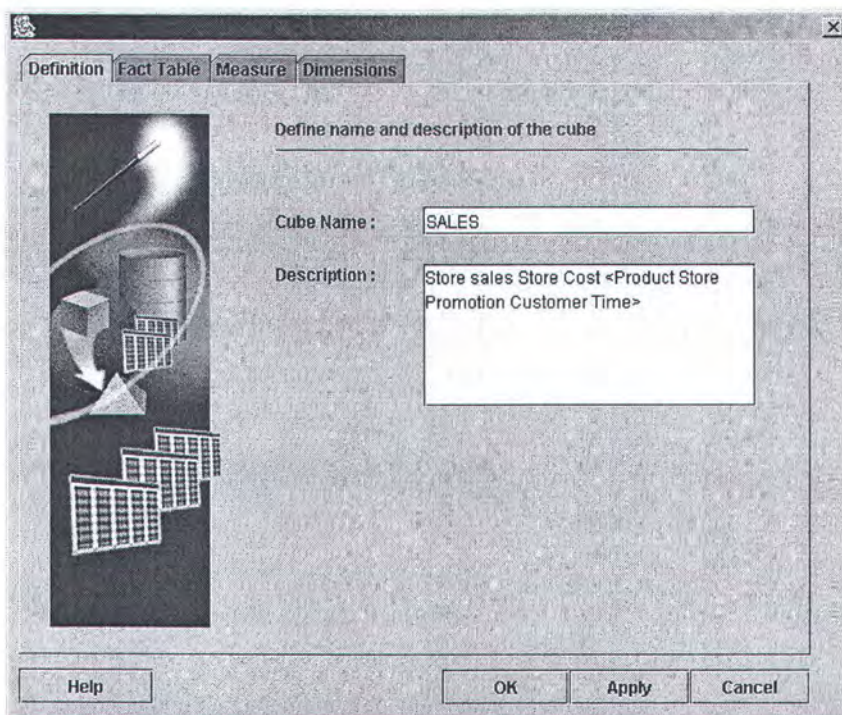




#### 4.4.7 FrameCube

Form *FrameCube* digunakan untuk membuat dan mengedit obyek *cube*. Form ini mengimplementasikan class *JTabbedPane* pada Java untuk membuat tab. Form ini memiliki 4 buah tab yaitu, Definition, Fact Table, Measures, dan Dimension. Masing-masing tab akan dijelaskan sebagai berikut.

##### 4.4.7.a FrameCube – Tab Definition

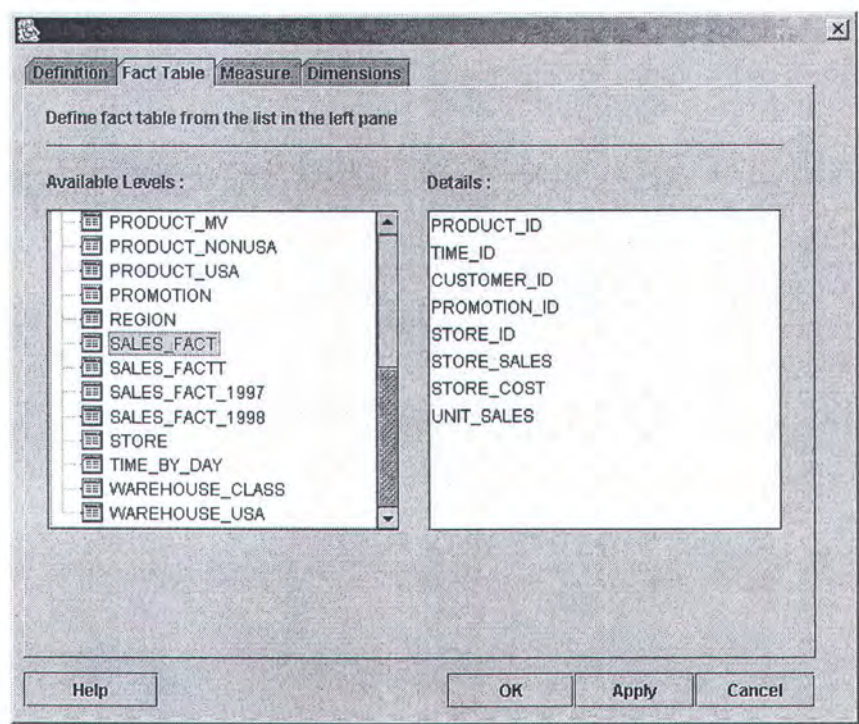


Gambar 4.18. Tab Definition pada FrameCube

Pada tab ini user mendefinisikan nama cube yang akan dibuat beserta deskripsi singkatnya. Tab ini mengimplementasikan class *JTextField* untuk menampung informasi nama cube. Selain itu, class *JTextArea* digunakan untuk menampung deskripsi cube.



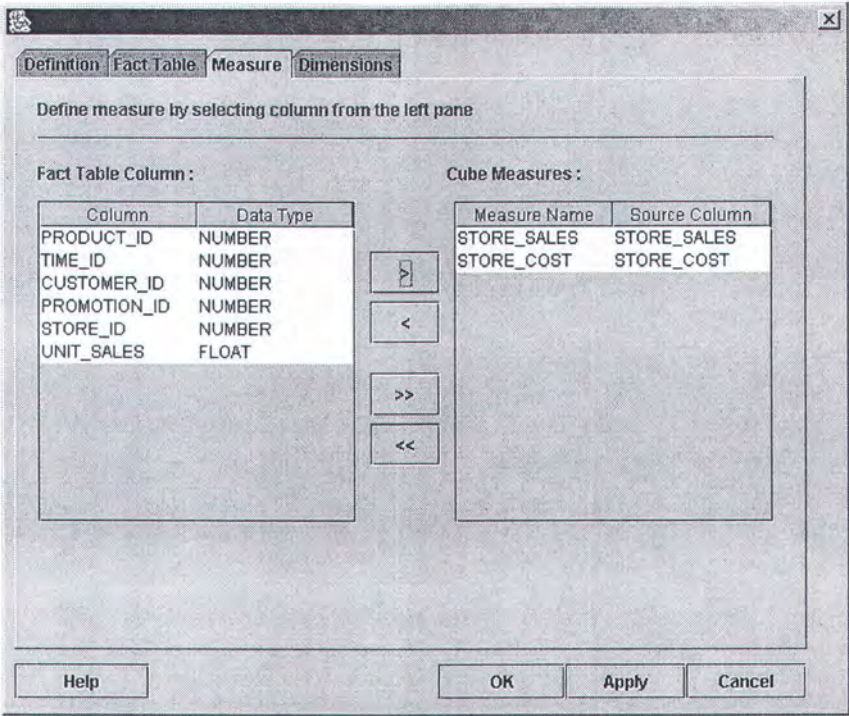
4.4.7.b FrameCube – Tab Fact Table



Gambar 4.19. Tab Fact Table pada FrameCube

Pada tab ini user menentukan tabel yang akan dijadikan tabel fakta pada cube. User memilih tabel pada panel bagian kiri, sementara panel kanan menunjukkan kolom yang dimiliki oleh tabel yang terpilih. Untuk mengimplementasikannya digunakan komponen *JTree* pada panel kiri, dan komponen *JList* pada panel kanan.

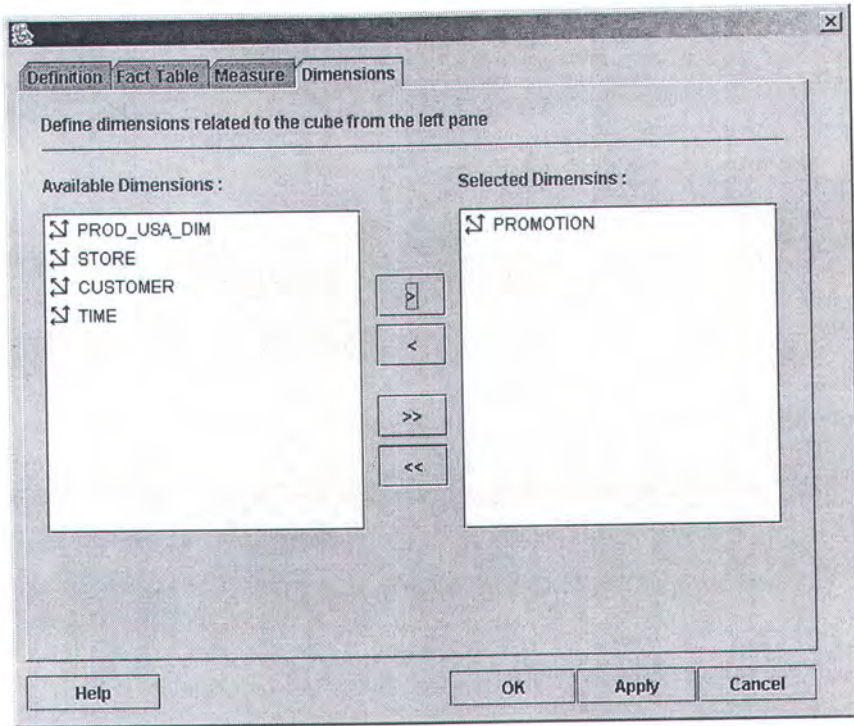
4.4.7.c FrameCube -Tab Measure



Gambar 4.20. Tab Measure pada FrameCube

Pada tab ini user menentukan kolom pada tabel fakta yang akan dijadikan *measure*. Sebagai implementasinya, digunakan dua buah komponen *JTable* untuk menampung informasi kolom dari tabel fakta disebelah kiri dan informasi kolom yang dipilih sebagai *measure* disebelah kanan.

#### 4.4.7.d FrameCube – Tab Dimensions



Gambar 4.21. Tab Dimension pada FrameCube

Pada tab ini user memilih dimensi mana saja pada daftar yang terlibat dalam pembuatan cube.

Untuk implementasinya digunakan dua komponen *JList* untuk menampung informasi dimensi yang tersedia dan dimensi yang terpilih. Serta empat komponen *JButton* untuk perpindahan posisi dimensi.



TO BELIEVE IS TO KNOW THAT A MAN IS NOT FINISHED  
WHEN HE IS DEFEATED. HE IS FINISHED WHEN HE QUILTS.

## BAB V

# UJI COBA DAN EVALUASI

## **BAB V**

### **UJICoba DAN EVALUASI**

Pembahasan dalam bab ini akan berkisar seputar uji coba yang telah dilaksanakan. Pertama akan dibahas tentang lingkungan tempat uji coba perangkat lunak ini. Yang kedua, penjelasan mengenai beberapa data yang digunakan dalam uji coba dan diakhiri dengan hasil uji coba dan evaluasi yang didapatkan.

#### **5.1 Lingkungan Uji Coba**

Spesifikasi komputer yang digunakan dalam rangka uji coba perangkat lunak ini adalah sebagai berikut :

- Prosesor : AMD Athlon 1,1 GHz
- Memori : 200 MB
- Sistem Operasi : Windows 2000 Professional Edition Service Pack 4.
- Server Database : Oracle 9i Database Release 2 (9.2.0.2.1) dengan OLAP option.

#### **5.2 Database Uji Coba**

Uji coba yang akan dilakukan menggunakan database Foodmart 2000. Database Foodmart merupakan suatu sistem database yang menangani penyimpanan dan pengelolaan informasi yang berkaitan dengan usaha pemasaran produk-produk makanan yang wilayah kerjanya di seluruh dunia. Database ini memiliki spesifikasi tabel sebagai berikut.



Table 5.1. Deskripsi tabel database foodmart

No	Nama Tabel	Keterangan
1.	ACCOUNT	Berisi data account pelanggan
2.	CATEGORY	Berisi data kategori
3.	CUSTOMER	Berisi data seluruh pelanggan
4.	EXPENSE_FACT	Berisi data pengeluaran
5.	INVENTORY_FACT	Berisi data penyimpanan produk
6.	PRODUCT	Berisi data produk
7.	PRODUCT_CLASS	Berisi data kategori produk
8.	PROMOTION	Berisi data promosi yang dilakukan
9.	REGION	Berisi data area pemasaran produk
10.	SALES_FACT	Berisi data penjualan
11.	STORE	Berisi data toko
12.	TIME_BY_DAY	Berisi data waktu
13.	WAREHOUSE_CLASS	Berisi data kategori gudang
14.	WAREHOUSE	Berisi data gudang

5.3 Ekstraksi Data

Foodmart merupakan database yang memiliki format Microsoft Access. Agar bisa digunakan dalam lingkungan database Oracle, maka harus dilakukan proses ekstraksi data terlebih dahulu. Ada banyak pilihan cara yang bisa digunakan untuk melakukan ekstraksi data, namun pada tugas akhir ini proses ekstraksi data dilakukan dengan menggunakan tool SQL Loader pada Oracle. Berikut ini adalah langkah-langkah yang harus dilakukan.

5.3.1 Konversi File Text

Tabel-tabel yang ada pada database Microsoft Access dikonversi terlebih dahulu menjadi file text. Untuk melakukannya, pada Microsoft Access memilih menu File > Export. Pada *window dialog* yang muncul mengisi nama file dengan tipe file .csv.



### 5.3.2 Membuat Control File

*Control* file digunakan sebagai informasi bagi database Oracle pada saat mengambil data pada proses ekstraksi. Untuk tiap tabel yang ada dibuat satu buah *control* file. Didalamnya, *control* file mendefinisikan sumber data (file dengan format .csv) dan tabel tujuan. *Control* file memiliki ekstensi .ctl. Berikut ini adalah contoh *control* file untuk tabel CATEGORY (category.ctl).

```
load data
  infile 'f:\category\category.csv'
  into table category
  fields terminated by ','
  ( category_id,category_parent,
    category_description,category_rollup )
```

### 5.3.3 Loading Data

Pada tahap terakhir, mengeksekusi perintah SQL Loader untuk me-*load* semua data ke dalam database Oracle. Perintah SQL Loader mengikuti aturan sebagai berikut

```
C:> SQLLDR userid = nama_user/password control =
nama_control_file
```

Misalnya,

```
C:> SQLLDR userid = anang/anang control = category.ctl
```

Proses loading data harus dilakukan untuk setiap tabel yang ada pada database Foodmart. Setelah mengeksekusi perintah diatas, maka data sudah di-*load* ke dalam database Oracle.

## 5.4 Skenario Uji Coba

Skenario yang akan diuji-cobakan pada perangkat lunak ini ada tiga. Skenario pertama, membuat metadata OLAP meliputi pembuatan dimensi standar, pembuatan dimensi waktu, pembuatan dimensi snowflake, serta pembuatan cube. Skenario kedua adalah mengedit metadata OLAP yang sudah dibuat pada skenario pertama. Skenario ketiga adalah melihat dan mengolah data multidimensi hasil skenario pertama dan kedua dalam bentuk *crosstab* dan grafik.

Untuk melihat apakah metadata OLAP yang dihasilkan oleh perangkat lunak ini valid digunakan dua program pembandingan, Oracle Enterprise Manager (OEM) dan `bi_checkconfig`.

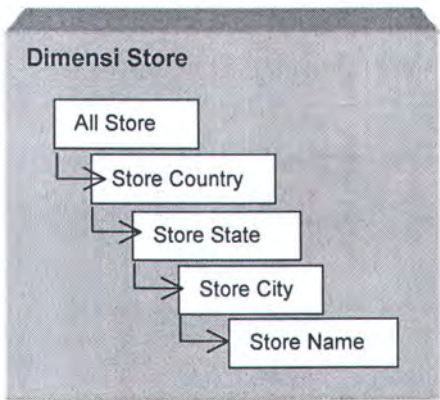
Oracle Enterprise Manager merupakan produk Oracle yang menyediakan manajemen sistem database secara terintegrasi. OEM memungkinkan user melakukan administrasi pada mode grafik tanpa harus berurusan dengan perintah-perintah SQL. OEM secara otomatis terinstal ketika database diinstal pada komputer.

Sedangkan program `bi_checkconfig` merupakan *utility* yang digunakan untuk memeriksa validitas metadata OLAP. Program ini bisa didapatkan (*download*) dari <http://otn.oracle.com/> , berupa `bi_checkconfig.zip`. Untuk penggunaannya, *copy*-kan dan unzip file ke home direktori dari Jdeveloper yaitu `JDEV_ORACLE_HOME/bibeans/bin`. File tersebut berisi `bi_checkconfig.bat` (Windows), `bi_checkconfig.csh` (UNIX), `BICheckConfig.class`, `BICheckConfig$1.class`, `BICheckConfig$2.class`, dan `bi_error_messages.html`.

5.3.1 Uji Coba 1 : Membuat Metadata OLAP

5.3.1.A Pembuatan Dimensi Standar

Untuk membuat dimensi, user mengklik kanan metadata dimension, kemudian mengklik *Create New Dimension* pada context menu yang muncul. Setelah itu akan muncul window pembuatan dimensi.



Gambar 5.1. Dimensi Store

Dimensi yang akan dibuat adalah dimensi STORE dengan spesifikasi seperti ditunjukkan pada gambar diatas. Dimensi STORE mengambil tabel STORE untuk dijadikan sebagai sumber datanya. Tabel STORE mempunyai spesifikasi data sebagai berikut :

Table 5.2. Tabel STORE

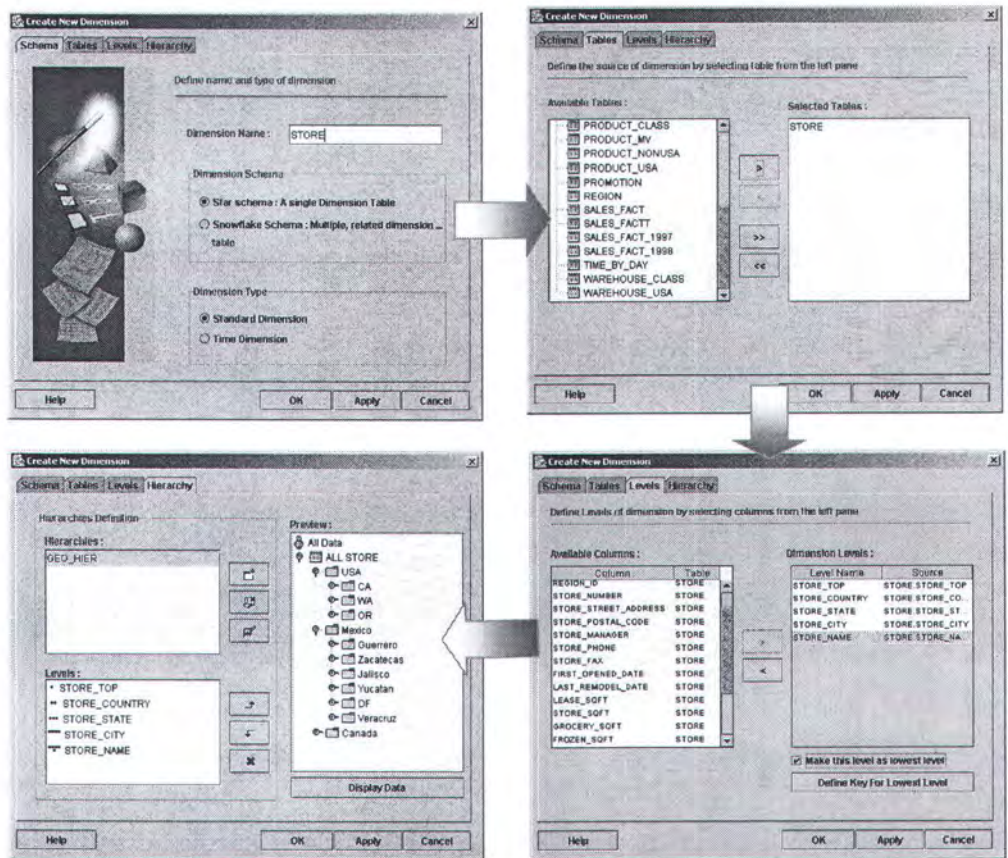
No	Kolom	Keterangan
1.	STORE_ID	Primary key
2.	STORE_TYPE	Tipe toko
3.	REGION_ID	Foreign key mengacu tabel Region
4.	STORE_NAME	Nama toko
5.	STORE_NUMBER	Nomer toko
6.	STORE_STREET_ADDRESS	Jalan toko
7.	STORE_CITY	Kota toko berada
8.	STORE_STATE	Lokasi state



9.	STORE_POSTAL_CODE	Kode pos toko
10.	STORE_COUNTRY	Lokasi negara
11.	STORE_MANAGER	Nama manager
12.	STORE_PHONE	Nomer telepon
13.	STORE_FAX	Nomer fax
14.	FIRST_OPENED_DATE	Tanggal pembukaan
15.	LAST_REMODEL_DATE	Tanggal terakhir perubahan
16.	LEASE_SQFT	Luas rak
17.	STORE_SQFT	Luas toko
18.	GROCERY_SQFT	Luas bagian grosir
19.	FROZEN_SQFT	Luas bagian pendingin
20.	MEAT_SQFT	Luas bagian daging
21.	COFFEE_BAR	Ada/tidak
22.	VIDEO_STORE	Ada/tidak
23.	SALAD_BAR	Ada/tidak
24.	PREPARED_FOOD	Ada/tidak
25.	STORE_TOP	Berisi “All Store”

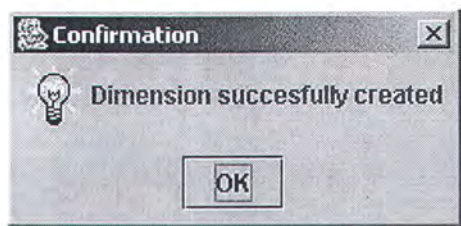
Untuk membuat dimensi STORE diperlukan langkah-langkah sebagai berikut.

1. Langkah 1, mengisikan nama dimensi (STORE) pada tab *schema*.
2. Langkah 2, menentukan tabel dimensi pada tab *tables*, dalam hal ini tabel yang dipilih adalah tabel STORE.
3. Langkah 3, menentukan level berdasarkan kolom yang ada pada daftar kolom. Ini dilakukan pada tab *levels*. Ada 5 kolom yang dipilih yaitu, STORE\_TOP, STORE\_COUNTRY, STORE\_STATE, STORE\_CITY, dan STORE\_NAME. Pada tab ini level STORE\_NAME dipilih sebagai *lowest level* dengan key STORE\_ID.
4. Langkah 4, membuat *hierarchy* dengan susunan level seperti yang diinginkan pada tab *hierarchy*.



Gambar 5.2. Pembuatan dimensi store

Setelah menekan tombol OK, maka akan muncul dialog seperti gambar dibawah yang menandakan bahwa proses pembuatan dimensi telah selesai dan sukses.

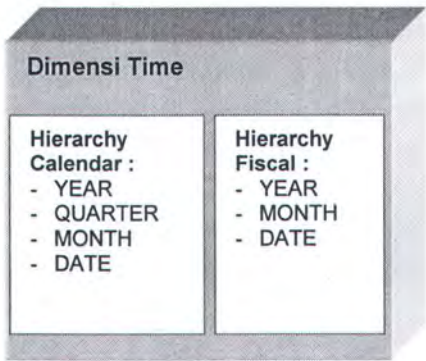


Gambar 5.3. Konfirmasi sukses



5.3.1.B Pembuatan Dimensi Waktu

Dimensi waktu memiliki spesifikasi yang berbeda dengan dimensi standar. Pada Oracle, dimensi waktu memiliki level-level yang sudah ditentukan. Level-level tersebut adalah Year, Quarter, Month, dan Date. Tidak hanya itu, hierarchy pada dimensi ini ada dua dan juga sudah ditentukan yaitu, *hierarchy* CALENDAR dan *hierarchy* FISCAL. Hal ini ditunjukkan pada gambar dibawah ini.



Gambar 5.4. Dimensi Time

Dimensi waktu pada uji coba ini akan mengambil tabel TIME\_BY\_DAY sebagai sumber datanya. Tabel TIME\_BY\_DAY mempunyai spesifikasi data sebagai berikut :

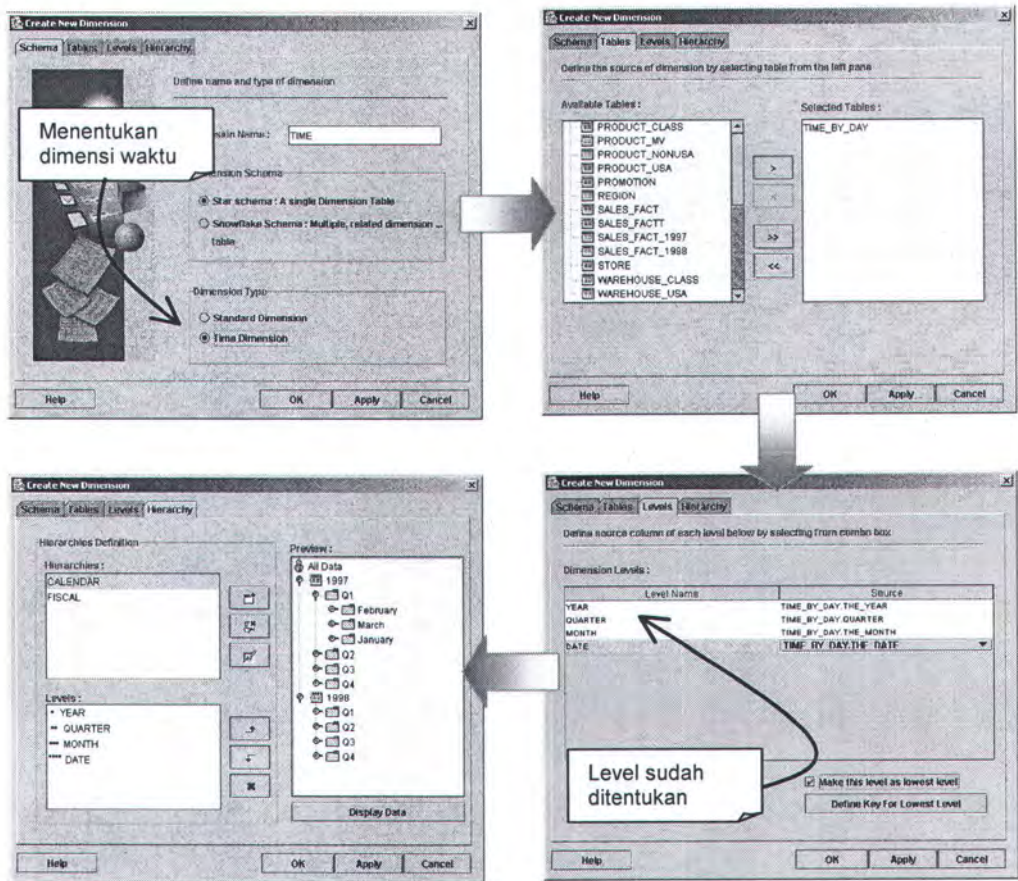
Tabel 5.3. Tabel TIME\_BY\_DAY

No	Kolom	Keterangan
1.	TIME_ID	Primary key
2.	THE_DATE	Tanggal
3.	THE_DAY	Hari
4.	THE_MONTH	Bulan
5.	THE_YEAR	Tahun
6.	DAY_OF_MONTH	Hari keberapa dalam bulan
7.	WEEK_OF_YEAR	Minggu keberapa dalam tahun
8.	MONTH_OF_YEAR	Bulan ke keberapa dalam tahun
9.	QUARTER	Triwulan
10.	FISCAL_PERIOD	



Untuk membuat dimensi waktu, diperlukan langkah sebagai berikut :

1. Langkah 1, tab *Schema*, mengisikan nama dimensi (TIME) dan memilih *radio button* Time Dimension pada bagian *Dimension type*.
2. Langkah 2, tab *Tables*, memilih tabel TIME\_BY\_DAY yang akan dijadikan sumber data.
3. Langkah 3, tab *Levels*, menentukan kolom pada masing-masing level yang sudah ditentukan oleh perangkat lunak. Pada langkah ini user memilih kolom dengan memilih pada komponen *combo box*. Berturut-turut kolom yang dipilih dari atas ke bawah adalah THE\_YEAR, QUARTER, THE\_MONTH dan THE\_DATE. Setelah itu, menentukan *key column* untuk level terendah (THE\_DATE) yang dalam hal ini dipilih kolom TIME\_ID karena merupakan *primary key* pada tabel TIME\_BY\_DAY.
4. Langkah 4, tab *Hierarchy*, perangkat lunak secara otomatis menentukan susunan struktur level pada masing-masing *hierarchy*. Pada langkah ini, tidak perlu dilakukan perubahan.

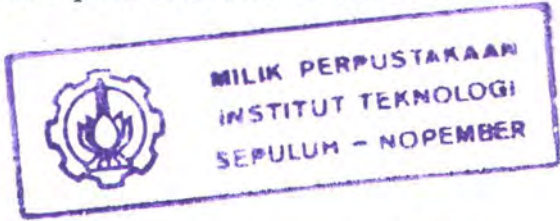


Gambar 5.5. Pembuatan Dimensi Time

Selanjutnya, menekan tombol OK, kemudian muncul pesan konfirmasi seperti pada gambar 5.3 yang menandakan bahwa proses pembuatan dimensi telah selesai dan sukses.

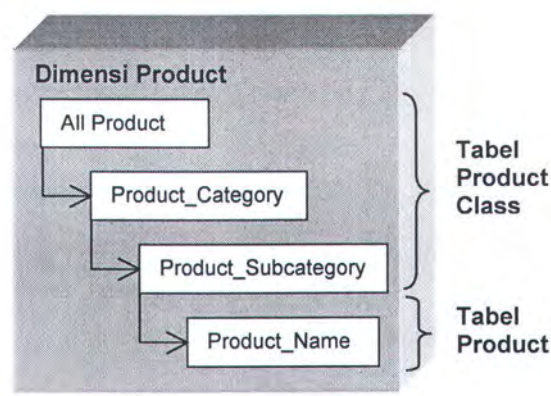
5.3.1.C Pembuatan Dimensi *Snowflake*

Dimensi *snowflake* adalah dimensi yang melibatkan lebih dari satu buah tabel. Pada uji coba ini akan dibuat dimensi PRODUCT dengan spesifikasi seperti ditunjukkan pada gambar 5.6. Terdapat 4 level yaitu, ALL\_PRODUCT, PRODUCT\_CATEGORY, PRODUCT\_SUBCATEGORY dan PRODUCT\_NAME. Tiga level teratas diambil datanya dari tabel





PRODUCT\_CLASS. Sedangkan level terbawah datanya diambil dari tabel PRODUCT.



Gambar 5.6. Dimensi Product

Untuk pembuatan dimensi ini digunakan dua tabel yaitu, PRODUCT dan PRODUCT\_CLASS. Masing-masing tabel memiliki spesfikasi data sebagai berikut :

Tabel 5.4. Tabel PRODUCT\_CLASS

No	Kolom	Keterangan
1.	PRODUCT_CLASS_ID	Primary key
2.	PRODUCT_SUBCATEGORY	Sub kategori produk
3.	PRODUCT_CATEGORY	Kategori produk
4.	PRODUCT_DEPARTMENT	Departemen
5.	PRODUCT_FAMILY	Keluarga produk
6.	PRODUCT_TOP	Berisi "All Produk"

Tabel 5.5. Tabel PRODUCT

No	Kolom	Keterangan
1.	PRODUCT_ID	Primary key
2.	PRODUCT_NAME	Nama product
3.	PRODUCT_CLASS_ID	Foreign key

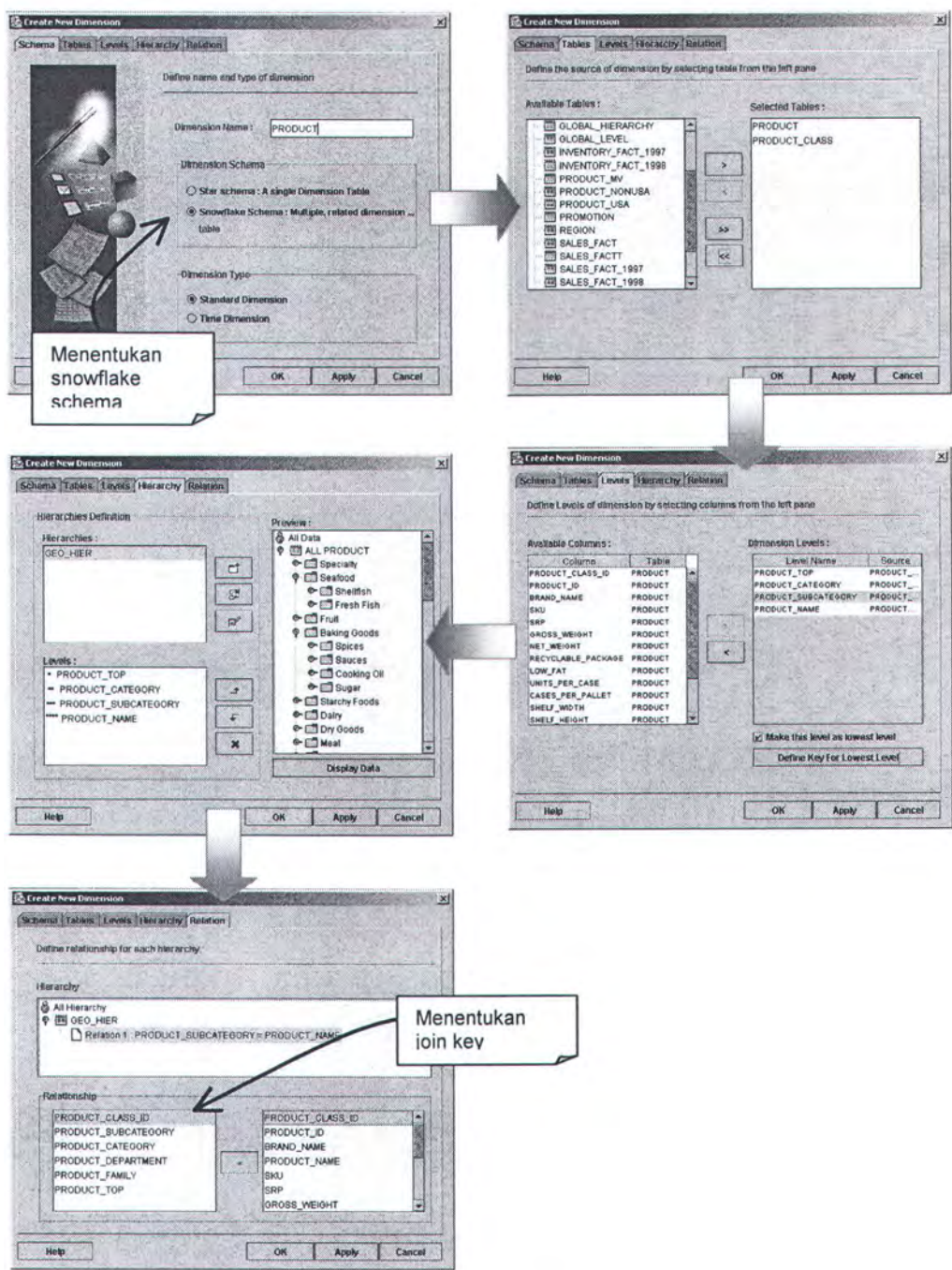


4.	BRAND_NAME	Nama brand
5.	SKU	
6.	SRP	
7.	GROSS_WEIGHT	
8.	NET_WEIGHT	
9.	LOW_FAT	Mengandung lemak/tidak
10.	UNITS_PER_CASE	Jumlah unit per bungkus
11.	CASES_PER_PALLET	
12.	SHELF_WIDTH	Lebar rak
13.	SHELF_HEIGHT	Tinggi rak
14.	SHELF_DEPTH	Kedalaman rak

Untuk membuat dimensi PRODUCT ini, langkah-langkah yang dilakukan adalah :

1. Langkah 1, tab *Schema*, mendefinisikan nama dimensi dan memilih *radio button* Snowflake Schema.
2. Langkah 2, tab *Tables*, memilih tabel PRODUCT dan PRODUCT\_CLASS dari daftar.
3. Langkah 3, tab *Levels*, memilih kolom yang akan dijadikan level. Kolom yang dipilih adalah ALL\_PRODUCT, PRODUCT\_CATEGORY, PRODUCT\_SUBCATEGORY, dan PRODUCT\_NAME. Kemudian menentukan *key* untuk level terendah. Dalam hal ini level terendah ada dua, yaitu PRODUCT\_SUBCATEGORY dengan *key* PRODUCT\_CLASS\_ID dan PRODUCT\_NAME dengan *key* PRODUCT\_ID. Setiap level terendah pada suatu tabel harus didefinisikan sebagai *lowest level*, dengan memberi tanda cek pada komponen *check box*.
4. Langkah 4, tab *Hierarchy*, menentukan susunan level sesuai dengan kebutuhan.

5. Langkah 5, tab *Relation*, menentukan *key join* untuk kedua tabel berdasarkan dua level yang berhubungan.

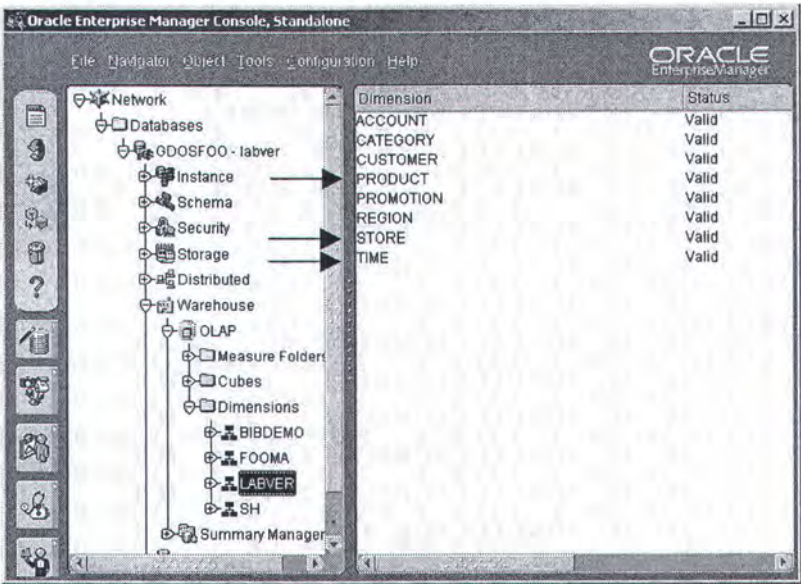


Gambar 5.7. Pembuatan Dimensi Product

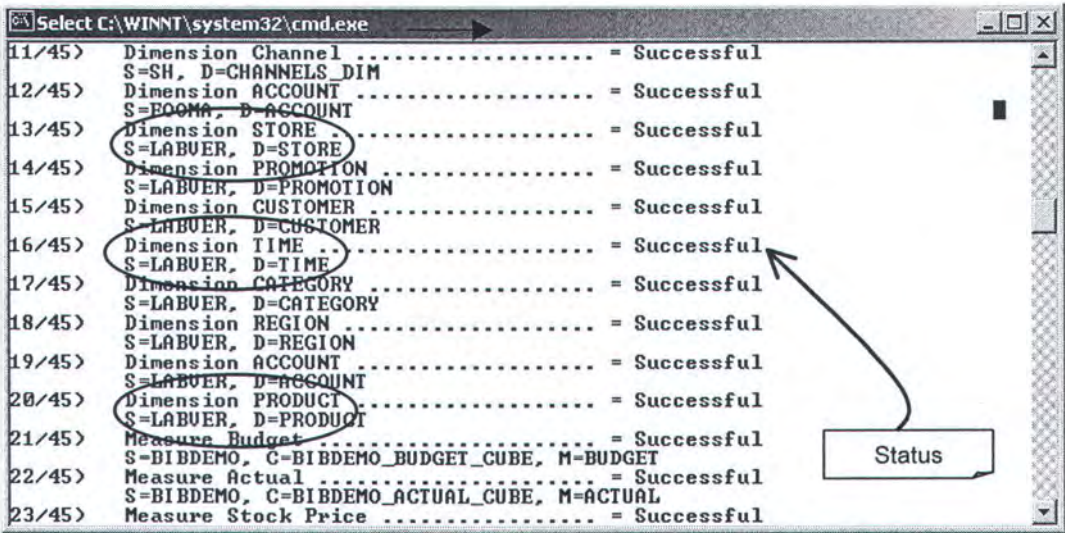


Selanjutnya, menekan tombol OK, kemudian muncul pesan konfirmasi seperti pada gambar 5.3 yang menandakan bahwa proses pembuatan dimensi telah selesai dan sukses.

Untuk membuktikan kevalidan metadata yang dihasilkan digunakan program Oracle Enterprise Manager dan bi\_checkconfig, seperti ditunjukkan gambar dibawah ini.



Gambar 5.8. (a) Hasil pembuatan Dimensi pada OEM



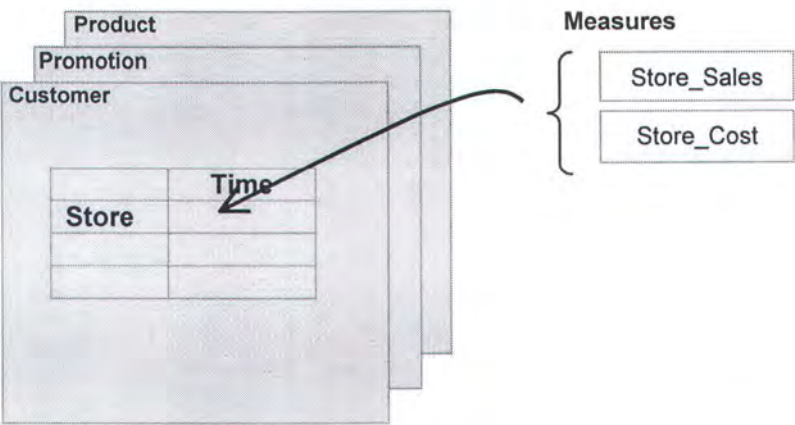
Gambar 5.8. (b) Hasil pembuatan dimensi pada bi\_checkconfig



5.3.1.D Pembuatan Cube

Untuk membuat cube, user mengklik kanan pada metadata *cube*, kemudian mengklik *Create New Cube* pada *context menu* yang muncul. Proses ini akan memunculkan *dialog* Create New Cube.

Pada uji coba ini, akan dibuat cube SALES dengan spesifikasi ditunjukkan pada gambar 5.8. Cube SALES melibatkan 5 dimensi yaitu PRODUCT, STORE, TIME, PROMOTION, dan CUSTOMER. Cube ini mempunyai *measure* STORE\_SALES, dan STORE\_COST.



Gambar 5.8. Cube Sales

Untuk membangun cube ini digunakan tabel SALES\_FACT sebagai tabel fakta. Berikut ini adalah spesifikasi data pada tabel SALES\_FACT.

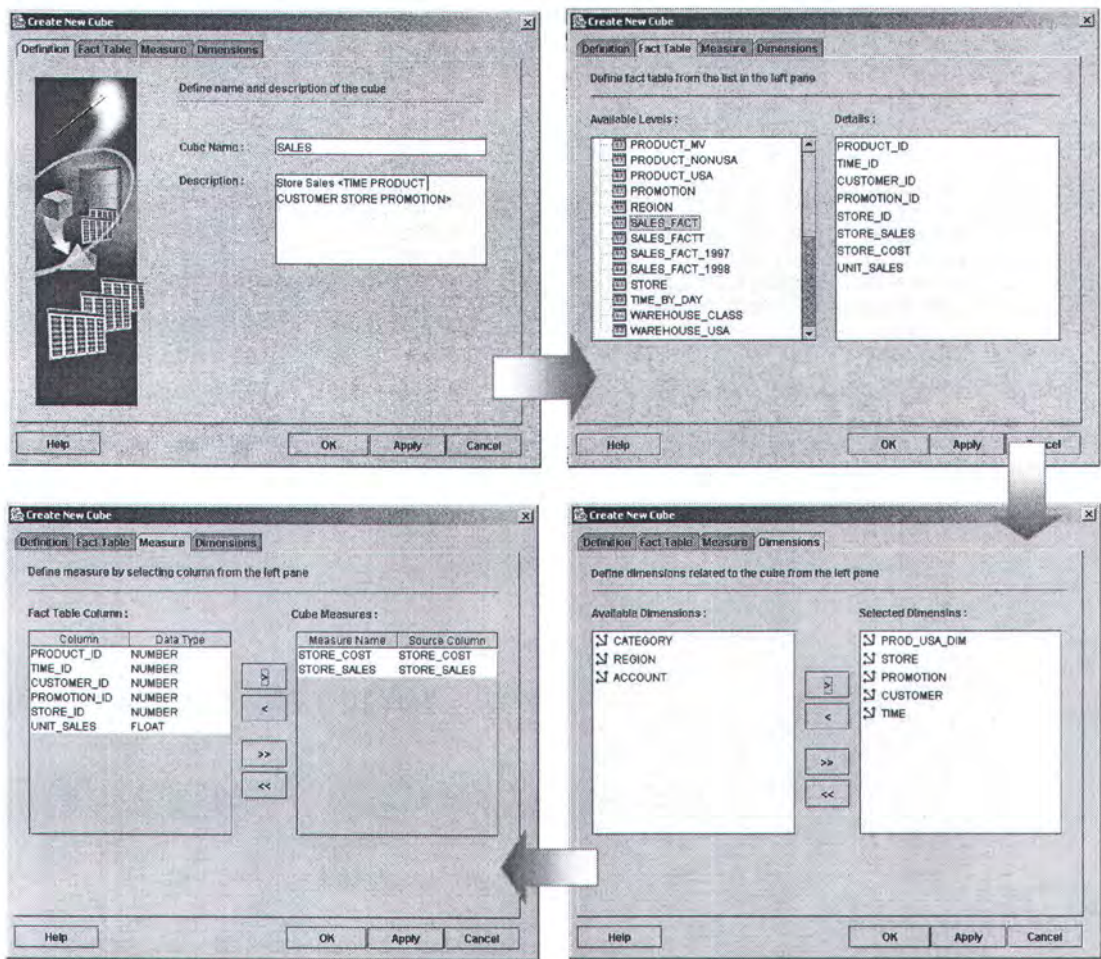
Tabel 5.5. Tabel SALES\_FACT

No	Kolom	Keterangan
1.	PRODUCT_ID	Foreign key tabel Product
2.	TIME_ID	Foreign key tabel Time_by_day
3.	CUSTOMER_ID	Foreign key tabel Customer
4.	PROMOTION_ID	Foreign key tabel Promotion
5.	STORE_ID	Foreign key tabel Store
6.	STORE_SALES	Measure

7.	STORE_COST	Measure
----	------------	---------

Langkah-langkah yang dilakukan untuk membangun *cube* ini adalah :

- 1. Langkah 1, tab *Definition*, mengisikan informasi berupa nama *cube* baru dan deskripsinya.
- 2. Langkah 2, tab *Fact Table*, memilih tabel SALES\_FACT pada daftar untuk dijadikan tabel fakta.

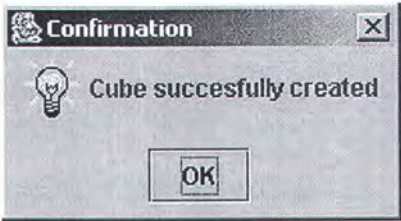


Gambar 5.9. Pembuatan Cube Sales



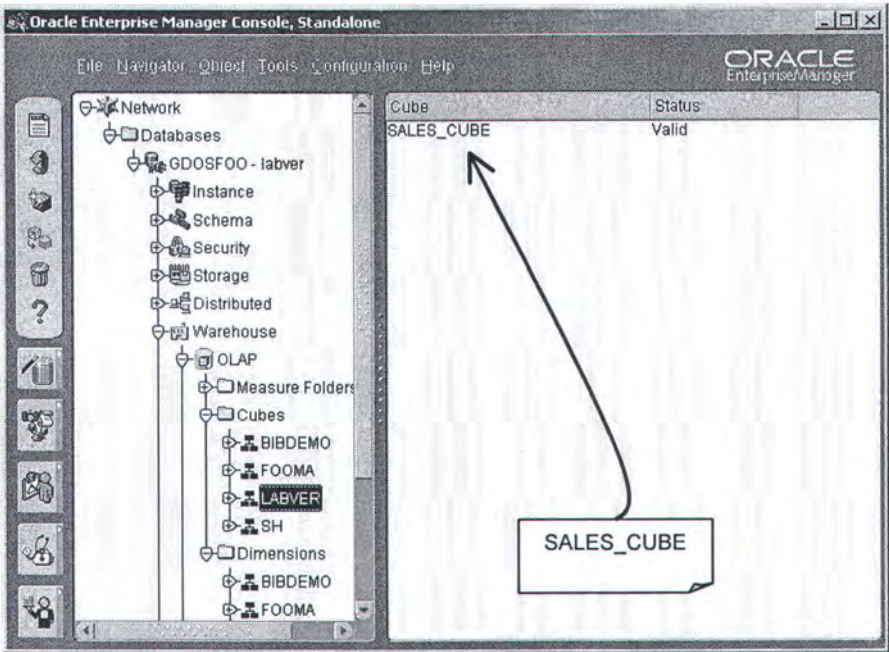
- 3. Langkah 3, tab *Measures*, memilih kolom yang akan dijadikan *measure*.  
Dalam hal ini yang dipilih adalah kolom STORE\_SALES dan STORE\_COST.
- 4. Langkah 4, tab *Dimensions*, memilih dimensi-dimensi yang terlibat dalam pembuatan *cube*.

Selanjutnya, menekan tombol OK, dan kemudian akan muncul *dialog* yang menyatakan bahwa proses pembuatan cube sudah selesai.



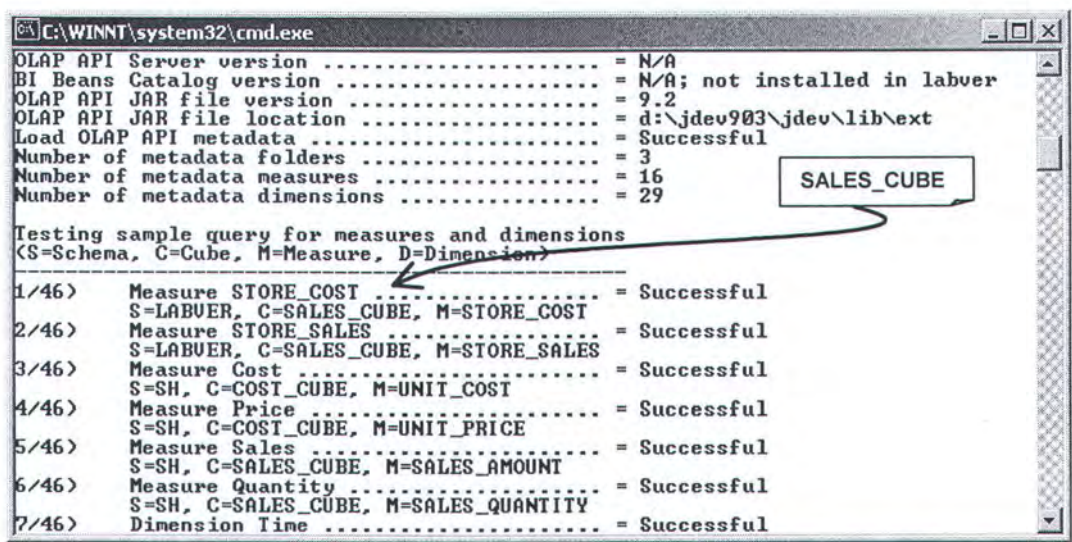
Gambar 5.10. Dialog konfirmasi cube

Jika hasil metadata dilihat dengan program OEM dan bi\_checkconfig, akan terlihat sebagai berikut.



Gambar 5.11. Sales\_Cube pada OEM





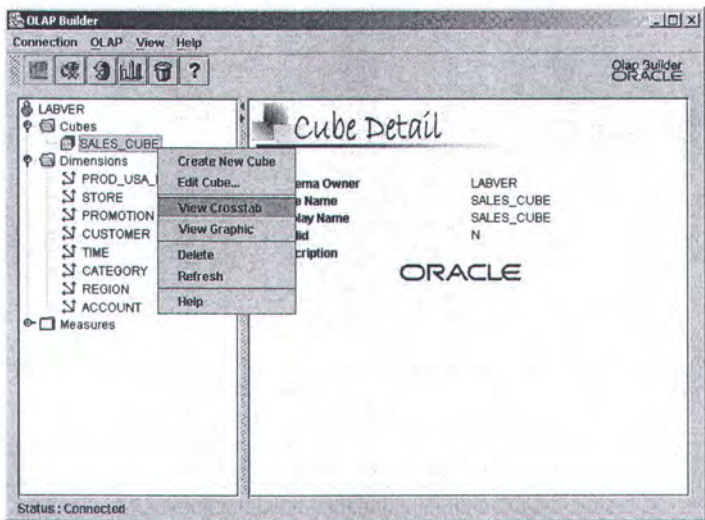
Gambar 5.12. Sales\_Cube pada bi\_checkconfig

5.3.2 Uji Coba 2 : Mengolah Data Multidimensi

Pada uji coba ketiga akan diperlihatkan proses *view* data multidimensi pada metadata yang telah dibuat sebelumnya. Juga akan dicoba untuk menampilkan data multidimensi dengan operasi-operasi OLAP seperti *Rotation/Pivoting*, *Slice and Dice*, dan lain-lain. *View* data menggunakan dua format yaitu, *Crosstab* dan Grafik.

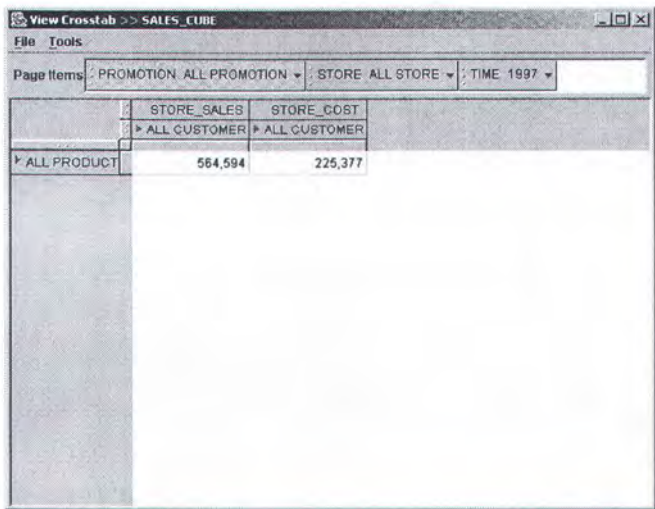
5.2.4.A SALES\_CUBE Crosstab

Untuk melihat data SALES\_CUBE dalam bentuk *crosstab*, dilakukan dengan cara mengklik pilihan *View Crosstab* pada *context* menu pada obyek SALES\_CUBE di *window* utama.



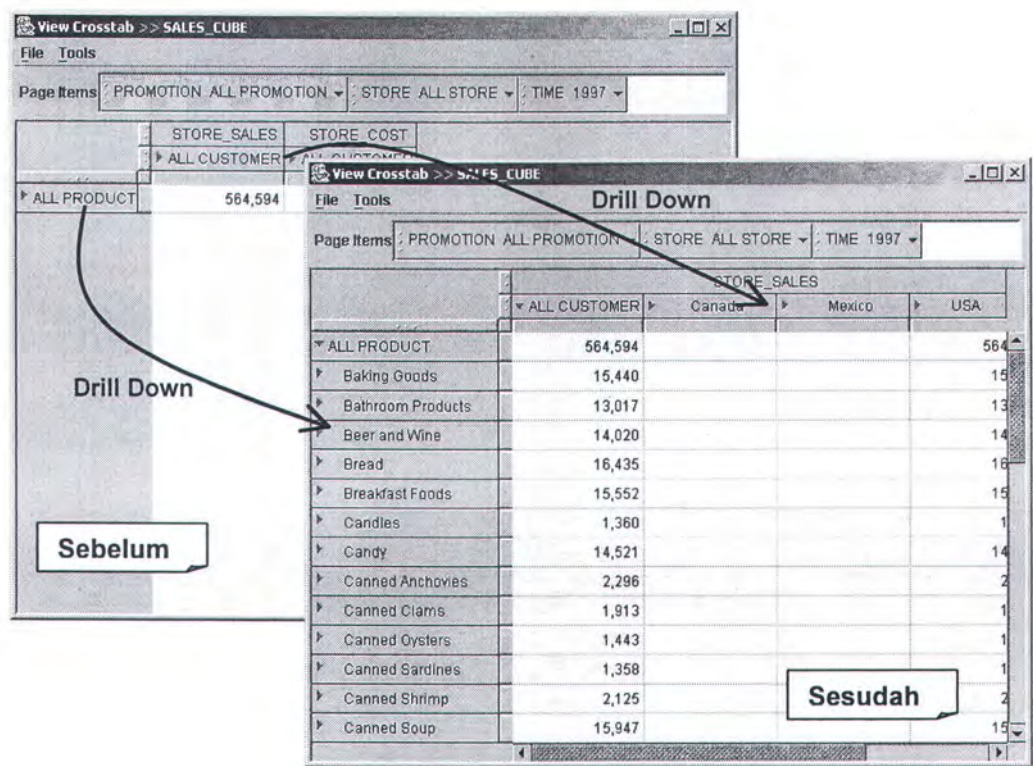
Gambar 5.10. Proses View Crosstab

Selanjutnya akan muncul *frame* sebagai berikut.



Gambar 5.11. Frame view crosstab

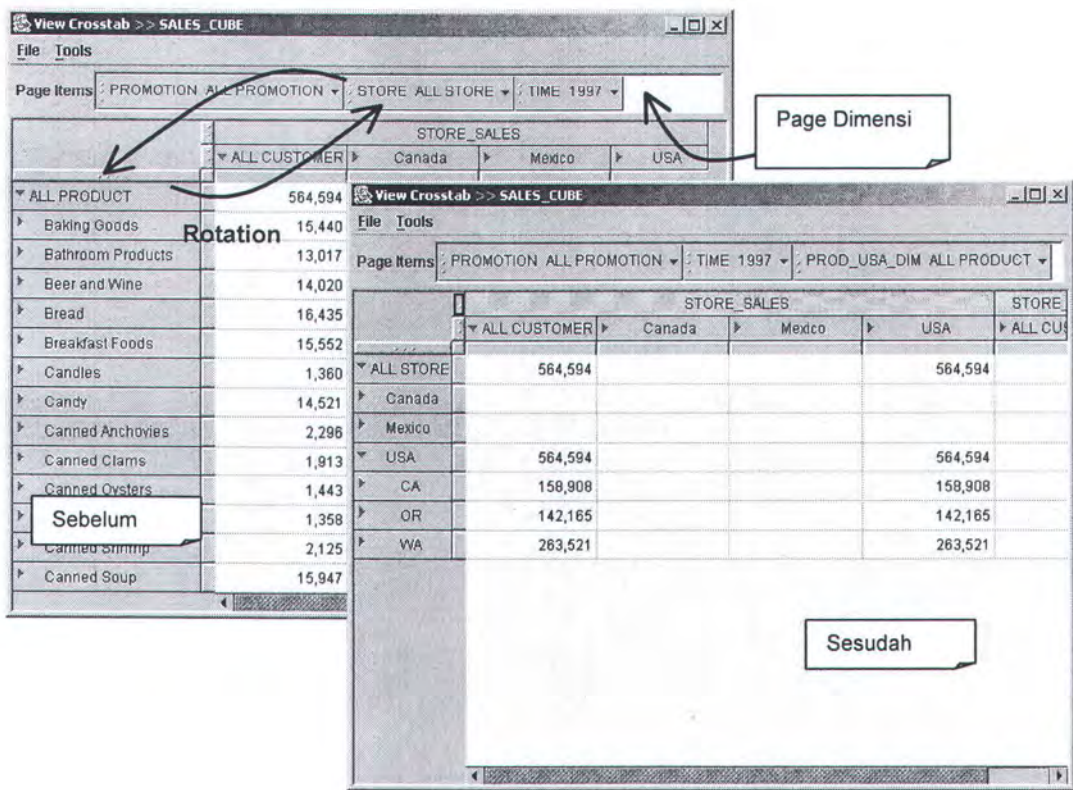
Pada *frame* View Crosstab dapat dilakukan operasi-operasi OLAP pada data yang ditampilkan. Untuk melakukan operasi *drill down*, cukup mengklik item *header* baris atau kolom pada *crosstab*. Gambar dibawah menunjukkan hasil operasi *drill down* pada dimensi PRODUCT dan dimensi CUSTOMER.



Gambar 5.12. Drill down

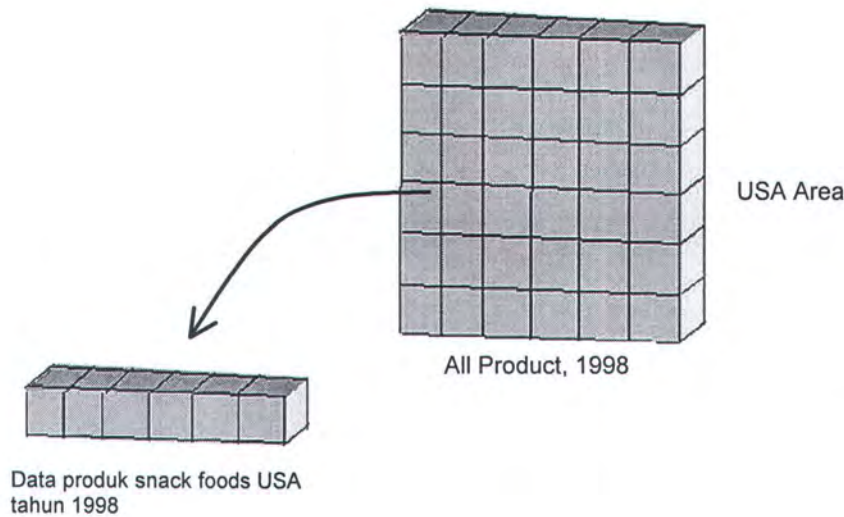
Untuk melakukan operasi *rotation/pivoting*, user memilih satu dari tiga dimensi pada *page*, kemudian men-*drag* dimensi tersebut ke *header* kolom atau baris. Selanjutnya men-*drag* dimensi yang sudah ada sebelumnya ke dalam *page*. Gambar dibawah ini menunjukkan operasi *rotation/pivoting* pada dimensi PRODUCT dan STORE.

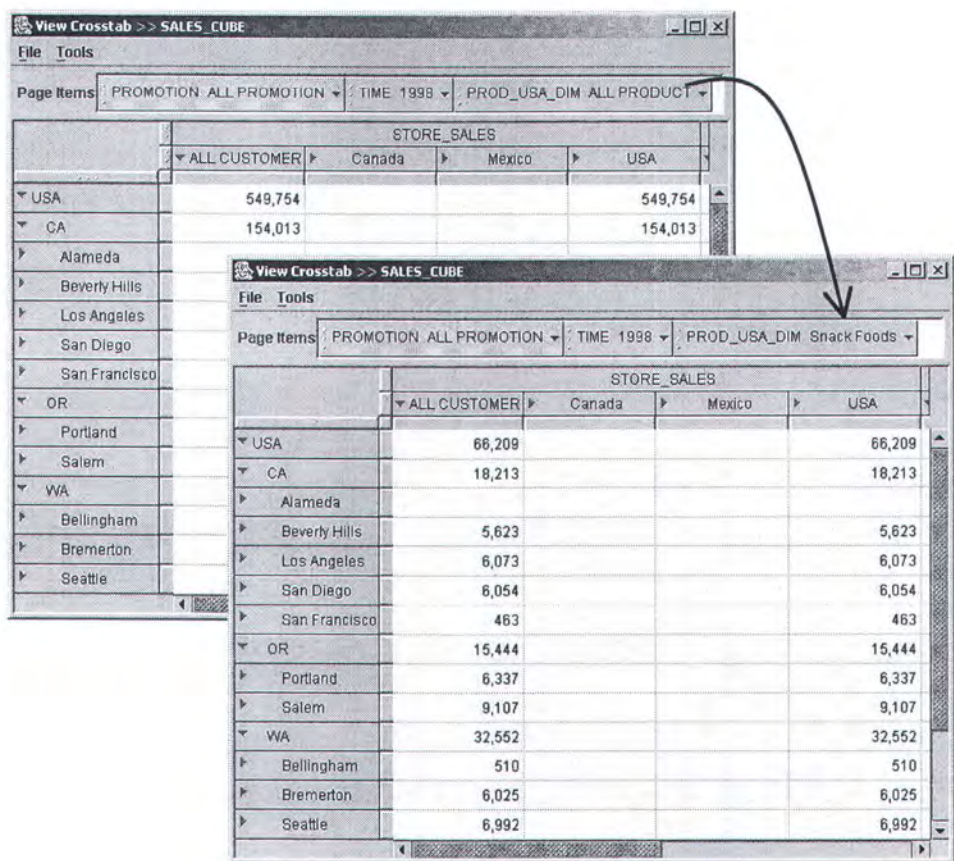




Gambar 5.13. Rotation/Pivoting

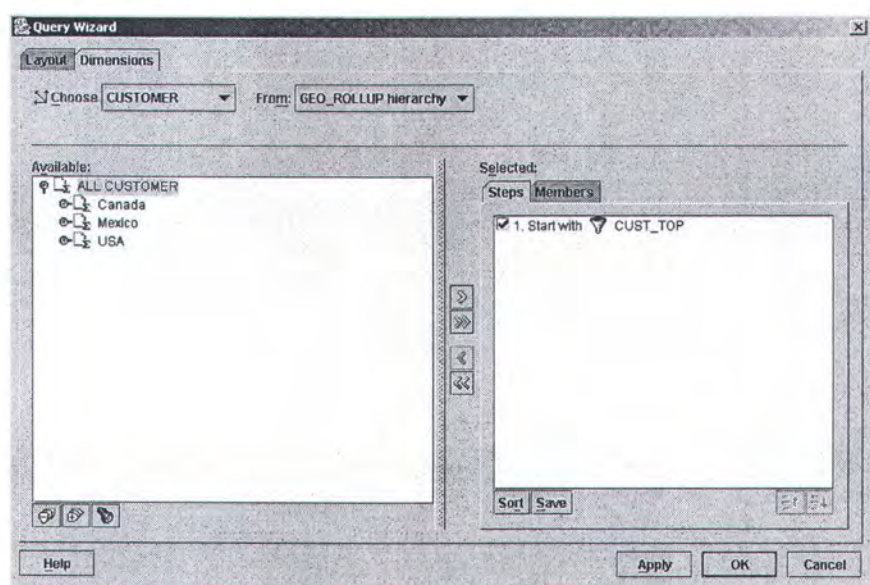
Selanjutnya, akan dilakukan operasi *slice and dice* pada data. Misalnya, manager ingin mendapatkan data penjualan semua produk dari berbagai lokasi di USA pada tahun 1998. Kemudian ingin mendetailkan lagi datanya hanya untuk produk





Gambar 5.14. Slice and Dice

Untuk melakukan pengolahan data lebih lanjut dapat dilakukan dengan memilih menu Tools > Modify Query pada *frame* View Crosstab.

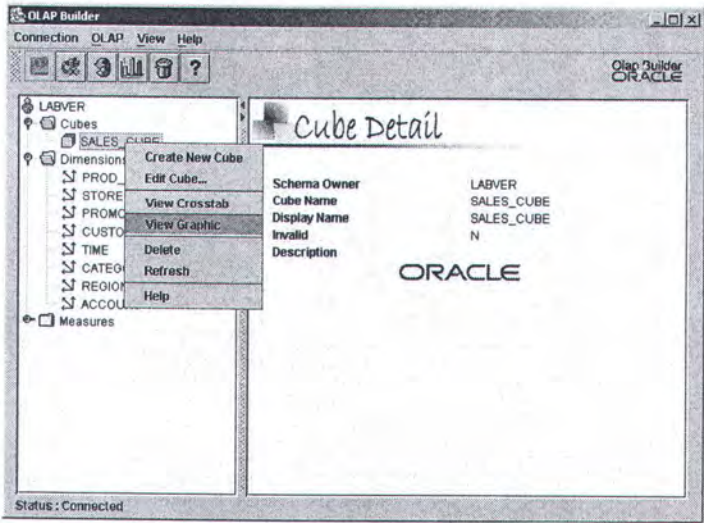


Gambar 5.15. Modify query



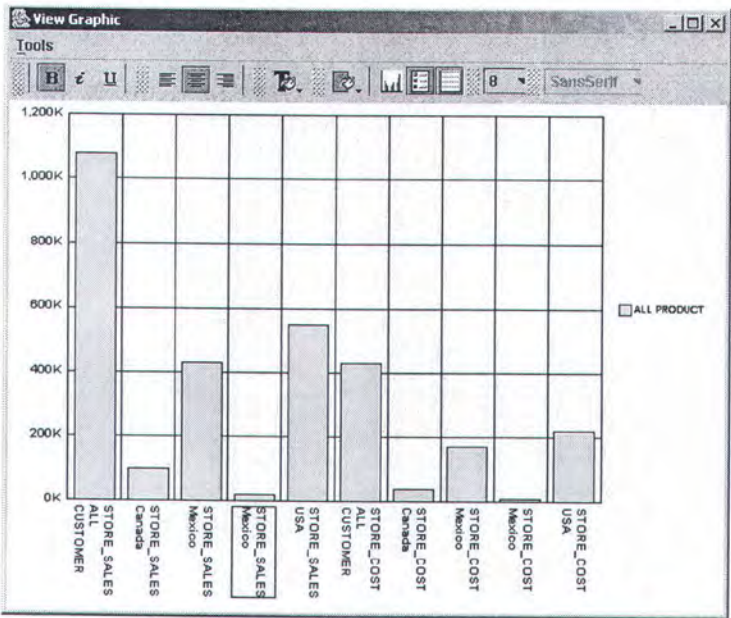
5.2.4.B SALES\_CUBE Grafik

Untuk melihat data SALES\_CUBE dalam bentuk grafik, dilakukan dengan cara mengklik pilihan *View Graphic* pada *context* menu pada obyek SALES\_CUBE di *window* utama.



Gambar 5.16. Cara melihat grafik

Selanjutnya akan muncul window berikut.



Gambar 5.17. Frame View Graphic



Pengolahan data pada grafik, pada dasarnya sama dengan *crosstab*, yaitu dengan memilih menu Tools > Modify Query pada *Frame View Graphic*.

5.4. Evaluasi

Dari hasil uji coba yang telah dilakukan, dapat dievaluasi beberapa hal yang berkaitan dengan perangkat lunak. Perangkat lunak mampu membuat metadata OLAP seperti dimensi standar, dimensi waktu dan dimensi snowflake dan *cube*. Dari hasil perbandingan dengan program lain, didapatkan bahwa hasil metadata OLAP yang dihasilkan perangkat lunak adalah valid. Perangkat lunak juga dapat digunakan untuk melihat dan mengolah data multidimensi dari metadata OLAP yang telah dibuat sebelumnya dalam bentuk *crosstab* dan grafik.

Berdasarkan hasil uji coba, maka dapat dibuat perbandingan proses pembuatan metadata antara OLAP Builder dengan perangkat lunak sejenis yaitu Oracle Enterprise Manager.

Tabel 5.6. Perbandingan Proses Pembuatan Metadata Antara OEM dan OLAP Builder

No.	Proses	OEM	OLAP Builder
1.	Menentukan attribute dimension	User (Manual)	Program
2.	Menentukan nama level dimension	Harus	Tidak harus
3.	Menentukan tabel sumber	Berkali-kali	Satu kali
4.	Preview data hierarchy	Tidak ada	Ada
5.	Menentukan nama hierarchy	Harus	Tidak harus
6.	Menentukan level terendah dan join key pada cube dan dimensi	User	Program
7.	Melakukan metadata refresh	User	Program

EVERY MORNING WE WAKE UP, WE HAVE TWENTY-FOUR  
BRAND NEW HOURS TO LIVE. WHAT A PRECIOUS GIFT!

IF ONLY WE BELIEVE.

BAB VI

PENUTUP



## **BAB VI**

### **PENUTUP**

Pada bab ini dijelaskan mengenai kesimpulan yang didapat setelah melakukan serangkaian uji coba. Dalam bab ini pula dijelaskan mengenai saran pengembangan perangkat lunak lebih lanjut.

#### **6.1 Kesimpulan**

Kesimpulan yang dapat diambil dari tugas akhir ini, dengan mengacu pada pembahasan bab-bab sebelumnya dan hasil uji coba yang telah dilakukan adalah sebagai berikut ini :

1. Pembuatan metadata OLAP dapat dilakukan dengan lebih mudah, dengan menggunakan aplikasi OLAP Builder. Hal ini dikarenakan dengan menggunakan OLAP Builder beberapa proses dan parameter dilakukan dan ditentukan secara otomatis oleh program. Tidak perlu pengetahuan tentang OLAP secara mendalam untuk memanfaatkan teknologi data warehouse dan OLAP.
2. Komponen Oracle BI Beans dapat digunakan bersama-sama dengan Oracle OLAP untuk membentuk aplikasi OLAP Builder yang digunakan untuk membentuk dan memanfaatkan metadata OLAP dalam proses analisis.
3. Aplikasi OLAP Builder terbukti mampu membuat metadata OLAP secara valid yang dibuktikan dengan Oracle Enterprise Manager.



## 6.2 Saran

Beberapa hal yang perlu dijadikan catatan dalam pengembangan sistem ini adalah :

1. Pengembangan perangkat lunak yang dapat mendukung operasi-operasi lintas program dan lintas database.
2. Penelitian lebih lanjut terhadap kelemahan teknologi OLAP pada Oracle yang tidak terdokumentasi serta perbaikan terhadap metadata OLAP yang tidak valid.

JUST BELIEVE TO THE ALMIGHTY. JUST BELIEVE TO THE  
MERCYFUL. JUST BELIEVE IN ALLH!. Laa Tahzan, Innallaha  
Ma'ana.....

## DAFTAR PUSTAKA

## DAFTAR PUSTAKA

- [PAU-02] Paul Lane. *Oracle9i for e-Business : Oracle9i Business Intelligence Beans* , Oracle Corporation, 2002
- [KEL-02] Kelly Chan. *Oracle9i Data Warehousing Guide, Release 2 (9.2), Part No. A96520-01*, Oracle Corporation, 2002
- [ORA-02] Oracle Corporation, "*Oracle9i OLAP User's Guide Release 2 (9.2.0.2), Part No. A95295-02*", 2002
- [API-02] Oracle Corporation, "*Oracle9i OLAP Developer's Guide to the OLAP API Release 2 (9.2), Part No. A95297-01*", 2002
- [DAM-03] Damayanti, Citra Dewi. *Perancangan Dan Pembuatan Aplikasi Analisa Data Warehouse Dalam Bentuk OLAP Berbasis Oracle 9i Dengan Menggunakan Oracle 9i Business Intelligence Beans Pada JSP (Studi Kasus Sistem Foodmart)*, Tugas Akhir, Jurusan Teknik Informatika, FTIF, ITS, 2002.



# LAMPIRAN



## LAMPIRAN A

### Tabel Kelemahan Teknologi OLAP Pada Oracle9i

(Disarikan dari forum OLAP pada <http://otn.oracle.com>)

No.	Kelemahan	Sebab
1.	BI Beans sudah terinstal, tapi tidak bisa melakukan koneksi ke Oracle OLAP.	JDBC driver tidak compatible  <b>Solusi :</b> Meng-copy 3 file pada direktori jdev903\bibbeans\jdbc\lib_92, yaitu : <ul style="list-style-type: none"> <li>- Classes12.jar</li> <li>- classes12dms.jar</li> <li>- nls_charset12.jar</li> </ul> ke direktori jdev903\jdbc\lib dan direktori jdev903\jdbc\lib_original.
2.	Setelah pembuatan metadata OLAP, status metadata menjadi tidak valid	Metadata belum direfresh  <b>Solusi :</b> Login sebagai OLAPSYS dan mengeksekusi perintah cwm2_olap_metadata_refresh.mr_refresh.
3.	Patch 9.2.0.2.1 dan BI Beans sudah terinstal, tapi belum bisa me-load OLAP API metadata.	Proses instalasi patch tidak tuntas.  <b>Solusi :</b> <ul style="list-style-type: none"> <li>- Login sebagai sysdba</li> <li>- Eksekusi \$Oracle_Home/rdbms/admin/catpatch.sql</li> <li>- Eksekusi \$Oracle_Home/rdbms/admin/utlrp.sql</li> <li>- Jalankan metadata refresh.</li> </ul>
4.	Jika suatu metadata yang dibuat tidak valid, maka semua cube yang ada pada schema tersebut menjadi tidak bisa dioperasikan	Tidak diketahui  <b>Solusi :</b> Menghapus metadata OLAP yang tidak valid.
5.	Metadata dimensi valid, tapi cube tidak dapat dioperasikan. Muncul pesan "vdb error..."	Ada dimensi yang tidak ditentukan hierarchy-nya.  <b>Solusi :</b> Menambahkan hierarchy pada dimensi yang tidak punya hierarchy

6.	Cube dapat dioperasikan, namun setelah reboot menjadi tidak valid dan semua cube dalam database juga tidak valid.	Tidak diketahui  <b>Solusi :</b> Mengeksekusi \$Oracle_Home/rdbms/admin/utlrp.sql, kemudian mengeksekusi metadata refresh.
----	---	---

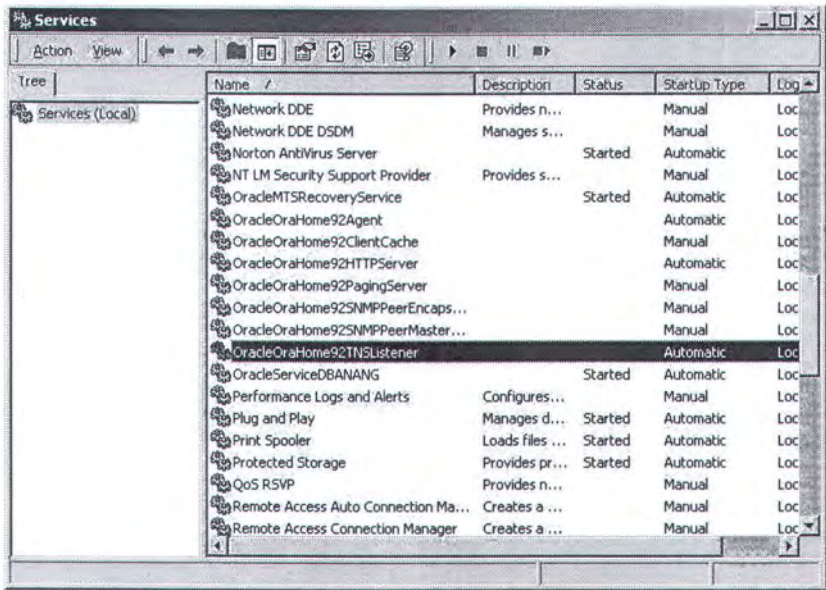


## Lampiran B

### PROSES INSTALASI PATCH 9.2.0.2.1 PADA ORACLE9I

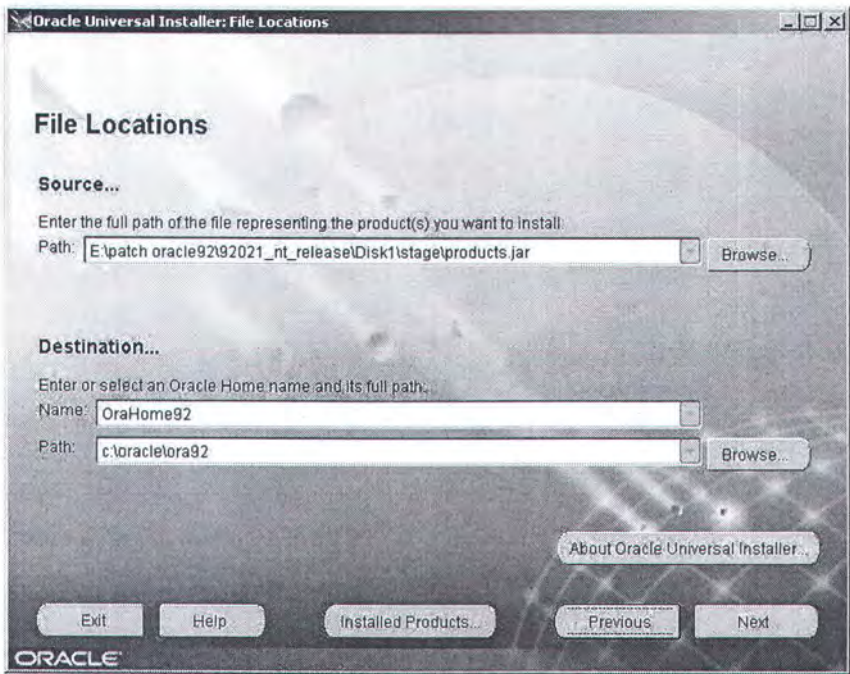
*Patch* 9.2.0.2.1 yang dirilis oleh Oracle ini berguna untuk memperbaiki kelemahan sistem terutama yang berkaitan dengan RDBMS, OLAP, JDBC, PL/SQL dan lain-lain. *Patch* ini bisa didapatkan pada website Oracle Metalink (<http://metalink.oracle.com>). Sebelum menginstall *patch* ini, komputer tujuan harus sudah terinstal Oracle9i Database release 2, karena *patch* ini hanya berfungsi untuk memperbaiki tidak untuk menginstal secara keseluruhan. Berikut ini adalah langkah-langkah yang harus dijalankan saat instalasi *patch* 9.2.0.2.1 pada Oracle 9i :

1. Login sqlplus sebagai sys as sysdba
2. Eksekusi perintah SQL berikut :  
  
SQL > drop public synonym xmlconcat;  
  
SQL > drop function xmlconcat;
3. Shutdown semua service yang sedang berjalan termasuk *instance* database dan tnslistener. Untuk menghentikan service dapat digunakan *administrative tool* pada sistem operasi.



Gambar B.1. Shutdown TNSListener

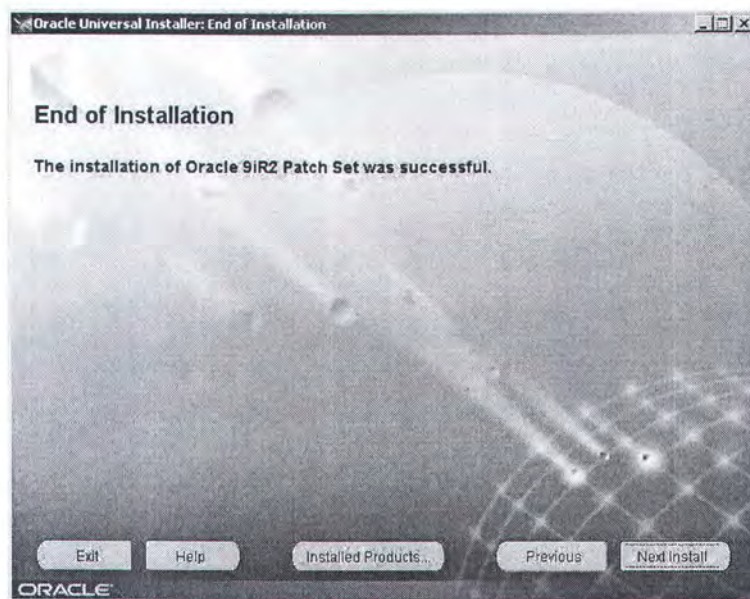
4. Selanjutnya, menjalankan *Oracle Universal Installer* dan memilih file yang akan diinstal. File *source* terletak pada direktori *stage* dengan nama file *products.jar*. Pada bagian *destination* dipilih Oracle\_Home tempat database yang akan diperbaiki.



Gambar B.2. Memilih source dan destination



5. Saat proses instalasi berjalan, beberapa program file akan diperbarui. Ada kemungkinan muncul pesan kesalahan karena program tidak bisa memperbarui file, hal ini disebabkan ada beberapa service yang masih berjalan. Jika hal itu terjadi, maka cari nama file yang dimaksud dan ubah nama file dimaksud.
6. Proses instalasi selesai ditandai dengan munculnya *window* dibawah ini.



Gambar B.3. Instalasi berhasil

7. Setelah instalasi selesai, hidupkan kembali semua service database dan TNSListener dengan cara yang sama ketika mematikannya.
8. Login sqlplus sebagai sys as sysdba
9. Eksekusi perintah berikut secara berturut-turut.

SQL > shutdown immediate

SQL> startup migrate

SQL> spool patch.log

SQL> @\$Oracle\_Home/rdbms/admin/catpatch.sql



```
SQL> spool off

SQL> shutdown

SQL> startup

SQL> @$Oracle_Home/rdbms/admin/utlrp.sql

SQL> exit
```

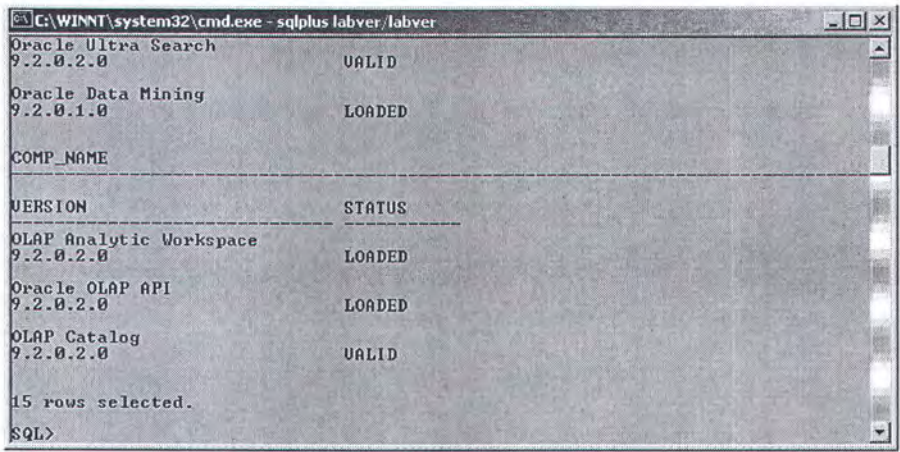
10. Untuk keperluan *refresh* metadata OLAP, user OLAPSYS harus di *unlock*.

Untuk melakukannya dapat menggunakan Oracle Enterprise Manager atau mengeksekusi perintah berikut ini :

```
SQL > alter user OLAPSYS account unlock
```

11. Periksa apakah OLAP catalog valid dengan cara mengeksekusi perintah berikut

```
SQL> select comp_name, version, status from dba_registry;
```



Gambar B.4. Cek komponen Oracle OLAP

12. Jika masih belum valid, login sqlplus sebagai OLAPSYS dan eksekusi perintah berikut

```
SQL> execute olapsys.cwm2_olap_installer.Validate_CWM2_Install;

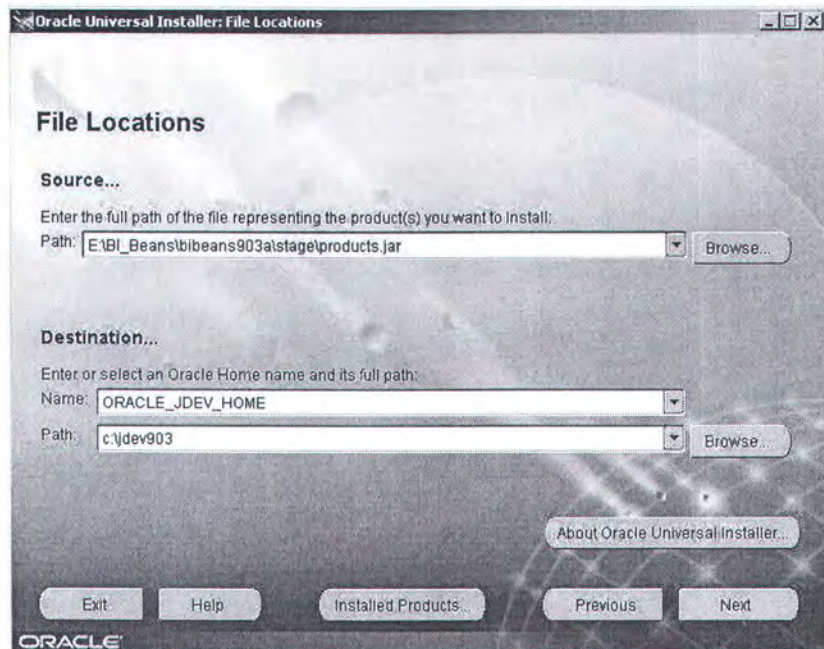
SQL> execute cwm2_olap_metadata_refresh.mr_refresh;
```

## Lampiran C

### PROSES INSTALASI BUSINESS INTELLIGENCE BEANS

BI Beans digunakan untuk database Oracle9i release 1 (9.0.1.3.x) dan release 2 (9.2.0.x.x). Proses instalasi kedua versi diatas berbeda. Yang akan dijelaskan berikut adalah proses instalasi untuk release 2. Untuk mendapatkan BI Beans bisa didownload dari Oracle Technology Network (<http://otn.oracle.com>)

1. Jalankan Oracle Universal Installer pada paket download (install\win32\setup.exe).
2. BI Beans harus diinstal pada Oracle Home yang mengacu ke direktori Oracle JDeveloper. File yang digunakan sebagai *source* bernama *products.jar* dan berada pada direktori *stage*.



Gambar C.1. Memilih source dan destination

3. Ikuti proses instalasi.

4. Setelah proses instalasi selesai, mengeset system variable pada komputer dengan dua variabel berikut.

Tabel C.1. Seting variabel

NAMA VARIABEL	NILAI
JDEV_ORACLE_HOME	Direktori dimana Oracle JDeveloper berada. Misalnya c:\jdev903
JAVA_HOME	Direktori dimana JDK berada. Misalnya c:\jdev903\jdk

5. Selanjutnya menginstal BI Beans *catalog* dengan cara mengeksekusi program `bi_installcatauto` yang berada pada direktori `bibeans\bin` dibawah direktori dimana Oracle JDeveloper berada, misalnya seperti contoh berikut ini.

```
C:\jdev903\bibeans\bin>bi_installcatauto localhost 1521 dbsisfo
```

Parameter :

localhost : komputer

1521 : Nomor port *service*

dbsisfo : Nama *service*

6. Proses instalasi selesai.



